DesignWorks™ 3.0 Release Notes

These notes provide supplementary information on DesignWorks version 3.0, released October 2, 1992.

Thank-you

Thank-you for purchasing the DesignWorks package from Capilano Computing. We are sure that you will find the package productive and enjoyable to use. Should you have any problems, questions, or comments, please contact us at the number given below.

What you have received

This package contains the 3.0 release of DesignWorks, and should contain the following:

1 Binder *
1 License Agreement
1 Release Notes
6 Program Diskettes
1 Reference Manual Package
1 Registration Card *
1 Support Plan Enrollment Card *

* These items will not be included in upgrade packages.

©1992 Capilano Computing Systems Ltd. All rights reserved.

Installation Notes

See also: Chapter II - Getting Started in the DesignWorks/Schematic manual.

In order to simplify installation and reduce the number of diskettes required to distribute DesignWorks, the release files are compressed onto a small number of diskettes using the Apple Installer and the StuffIt™ utility.

The Installer allows you to install the entire system in one operation, or to choose specific groups of items to be installed.

Program Installation Procedure

Before starting the installation procedure you should take the following preparatory steps:

1) The Installer requires that it be the only program running on the machine it is installing onto. Before starting the Installer, shut down all other applications.

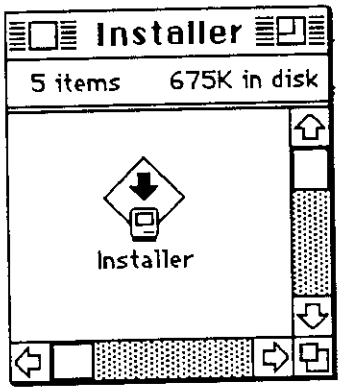
2) If you have any virus detection software on your machine, you may find some interaction with the Installer. It is best to disable any virus software before proceeding.

3) Depending on which disk you are installing on, the Installer may require that you disable File Sharing, or at least ensure that no folders are exported from your machine during the installation.

4) A complete installation of DesignWorks and all its libraries requires close to 30 megabytes of disk space. Most of this space is taken up by parts libraries. If you want to install the whole system, you will need to ensure that this amount of space is available. You can use the "Customize" option to choose only a basic program installation to get started. In any case, a minimal installation requires at least 4 megabytes.

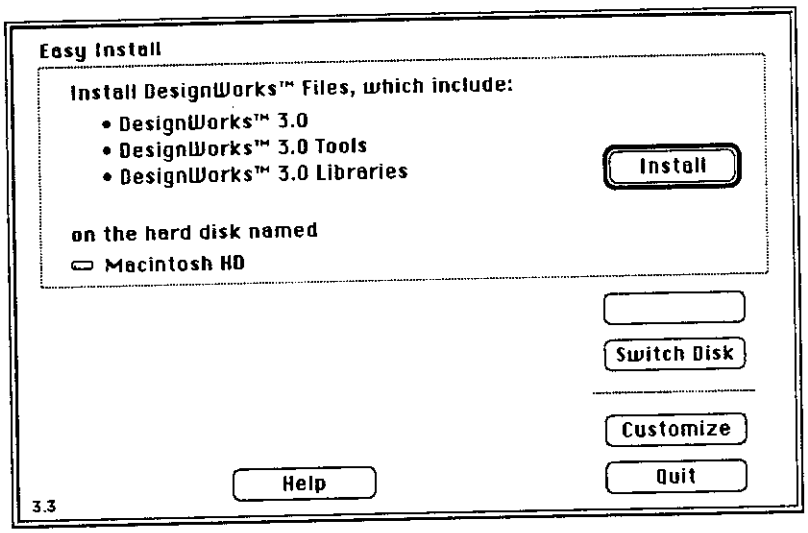
To start the installation:

1) Insert the diskette labeled "Installer". You should see the following files on the diskette:



2) Do not copy any of these files onto your hard disk! Double-click on the file "Installer". This will start up the installer program.

3) The following box will be displayed:



At this point you can choose one of two paths:

- Clicking the Install button will install all the DesignWorks files on your disk.
- Clicking the Customize button will allow you to choose which library groups you install. If you are installing DesignWorks 3.0 for

the first time, you may wish to install only the basic program set, since this is more than sufficient to get started with.

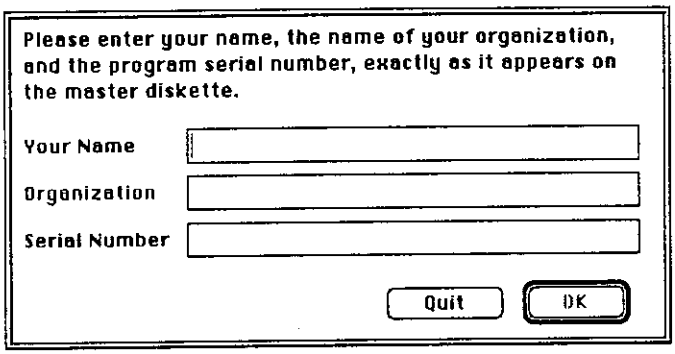
5) The installer will now copy the files from the installation disks, and then unpack the files onto your disk. It will ask for the diskettes it requires as it proceeds.

6) The program can now be started by clicking on the "DesignWorks™ 3.0" icon inside the DesignWorks 3.0 folder!

A basic set of libraries are included in the Basic Libs folder in the program installation. Before proceeding with a significant design, you should review the Guide to Libraries in the Notes section of the manual.

Serial Number Entry

The first time you start the program, you will see the following box:


Please enter your name, the name of your organization, and the program serial number, exactly as it appears on the master diskette.
Your Name
Organization
Serial Number
Quit
OK

Enter the following information:

- Your name and the name of your organization. This information will be displayed each time the program is started.
- The serial number as provided with your master diskette. This is a sequence of 20 letters and numbers. It can be entered either with or without separating spaces. All letters must be upper case.

After this information is entered once, it is kept with the program and you will not need to enter it again.

DesignWorks uses a network copy protection scheme that limits the number of users of a single copy on a network. The serial number contains encoded in it the number of authorized users and validation codes for add-on tools. You can upgrade your copy at any time by contacting Capilano Computing at the phone number given elsewhere in this manual.

Correcting or Re-entering the Serial Number

If for any reason you should wish to re-enter this information, you can hold the Option key pressed while starting the program from the Finder. The program will then redisplay the serial number entry box.

Troubleshooting Installation

PROBLEM: Program seems to run, but no windows appear and the Window and Tools menus have nothing in them.

CAUSE: The program has failed to locate any of the tool files. This could be due to any of the following:

- The tools are not in the same folder as the DesignWorks™ 3.0 application or the Tools folder has been renamed.
- The DesignWorks Setup file has been renamed or is not in the same folder as the DesignWorks™ 3.0 application.
- The folder name specified in the TOOLFOLDER line of the DesignWorks Setup file is incorrect.

PROBLEM: Error message about "Could not open library XXXX"

CAUSE: The name or location of a library specified in the DesignWorks Setup file is incorrect.

PROBLEM: Error message about "Unexpected changes have been made to the application..." appears on startup.

CAUSE: The program checks itself for unexpected changes each time it starts up, as a precaution against viruses. If you see this message, you should run a virus checking program on your disk and re-install the program from the master diskettes (being careful to keep them write-protected). In most cases, you can continue and execute the program without major ill effects, but there is a risk that you will contaminate or corrupt other files in your system.

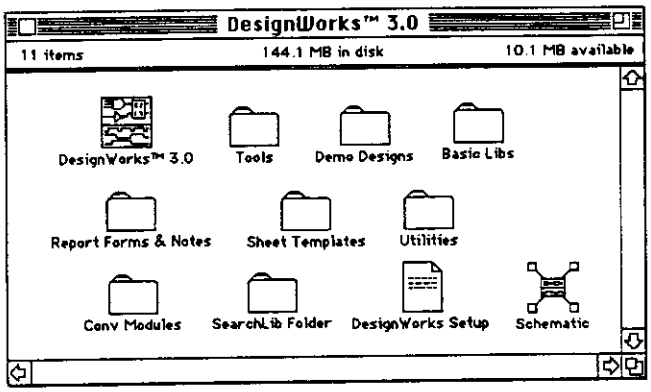
PROBLEM: Error message about "Unable to allocate memory or bad format for module XXXX".

CAUSE: This message can appear for one of the following reasons:

- The module in question is obsolete, i.e. it is part of an older DesignWorks release.
- The module file is corrupted.
- There was insufficient memory to load the module.
- The module was not authorized for operation with your copy of DesignWorks (in this case another message will precede this one).

Files Installed

The following Finder window shows a typical DesignWorks installation:



DesignWorks™ 3.0	This is the program itself. Double-click on this icon to start it up. This program is only a "shell" which loads the Schematic file and other tools.
Demo Designs	This is a folder containing some sample design files.
DesignWorks Setup	This is a text file which determines where DesignWorks searches for modules and libraries, default colors for objects, an initial design file to open, and other parameters. These are described in Appendix D of the DesignWorks manual. The basic setup file provided assumes the existence of a folder called Tools containing at least the "LibIO" module, and a folder called "Basic Libs" containing libraries.
Tools	This is a folder containing any code modules that are to be loaded when DesignWorks starts up. The name of this folder is specified in the Setup file. If the folder name is changed, as corresponding change must be made in the Setup file or no modules will be found. The only module that is essential for normal DesignWorks operation is "LibIO", which handles the Parts palette and all device library operations.
Sheet Templates	This folder contains template files for standard sheet sizes. These can be imported to your design using the Import Sheet Info command.
Report Forms & Notes	This is a folder containing the "form" files and circuit examples for the various standard report formats produced by the Report module. Information on these formats is given in the Report Generator manual and in text comments in the form files themselves.

Basic Libs	Any library files in this folder will be opened automatically when the program is started.
Libs	This folder contains all libraries installed by the Installer program. These libraries are not opened automatically at startup, but can be opened using the Open Lib command.
Utilities	This folder contains a number of utility tools and associated documentation.
Conv Tools	This folder contains the tools needed for conversion of pre-3.0 DesignWorks files to the new format. See the notes below before proceeding.
SearchLib Folder	This folder contains items for the SearchLib tool, which is used to locate parts in the libraries by description.
Schematic	This is the main schematic drawing tool for DesignWorks. It is loaded automatically when DesignWorks starts up. The program should normally be started by double-clicking on the DesignWorks™ 3.0 icon rather than on this file.

If you are updating from an older version of DesignWorks

If this package is an upgrade for an existing DesignWorks installation, please note that a conversion procedure must be performed on any files created using version 2.5.6 or older. Instructions for this procedure are in the DesignWorks/Converter manual behind the Notes tab in the binder.

DesignWorks 3.0 uses a different file format than all previous versions and will not directly read old files. You will need to use the Converter and ConvLib tools and possibly make some manual changes to your files.

Site or Network Licensing

When you purchase a copy of our software you are licensed to store and use it on a single computer at one time, making copies only for backup purposes. We have very attractive multiple unit licensing arrangements for organizations that are interested in using our software on a number of machines on the same network. Please contact us directly for more information.

Copy Protection

For your convenience, the DesignWorks disk is not copy protected in any way, allowing you to copy the program onto your hard disk or backup floppies as desired. This does **not** imply that you are allowed to make copies for use on more than one computer, in fact you are specifically prohibited from doing so by your license agreement. By respecting that agreement, you will help us continue to bring you useful software products at reasonable prices.

Technical Support

If you have any questions, problems, suggestions, congratulations or whatever, you can contact us by phone at (604) 522-6200 (9 - 5 Pacific time, please ask for DesignWorks Technical Support), by fax at (604) 522-3972, on AppleLink at D0816 or by mail at:

Technical Support,
Capilano Computing,
960 Quayside Drive, Suite 406,
New Westminster, B. C.
Canada V3M 6G2

Bulletin Board Service

The DesignBoard BBS is available especially for our customers to access technical support, download updates and technical notes, exchange files and communicate with Capilano personnel.

Access number: (604) 522-4361

Access parameters: 2400 Baud, 8 bits, no parity

On your first access you can use basic functions and request validation, or you can phone us in advance and we will create an ID for you.

Credits

This program is being distributed using the Apple Installer version 3.3 licensed from Apple Computer, Inc. and *StuffIt* installer technology licensed from Aladdin System. *StuffIt* is a file compression, encryption and file archival utility by Aladdin Systems, 217 E. 86th St., #153, NY, NY, USA.

DesignWorks/Schematic

Version 3.0

Schematic Drawing Tool

for the Apple Macintosh™

Reference Manual

August 26, 1992

406 - 960 Quayside Drive
New Westminster, B.C., Canada, V3M 6G2
(604) 522-6200

©1992 Capilano Computing Systems Ltd.

Table of Contents

Chapter I - Introduction 1

Where to Start 1
Chapter Organization 2
New Schematic Features in 3.0 3
Hierarchy 3
Attributes 3
Sheet Templates 4
Packaging 4
Browser Tool 4
Other Features 5
Interface Features 6
Limitations In This Version 7
Compatibility 7
Introduction to MEDA™ 8
The MEDA Concept 8
Creating MEDA Modules 10
Notes Regarding Copyright 10

Chapter II - Getting Started 11

Installation 11
General Rules 11
External Tools 12
Default Library Folder 12
Serial Number Entry 12
Troubleshooting Installation 13
File Compatibility 14
Memory Usage 15
The Setup File 15

Chapter III - User Interface 17

Mouse Button Usage 17
Dialog Boxes 17
Window Usage 18
Circuit Windows 18
Tool Windows 18
Closing a window 18
Redisplaying a window 19
Resizing a window 19

Moving a window 19
The Window Menu 19
Keyboard Usage 19
Pop-up Menus 20
Tool Palette and Status Display 20
Moving the Tool Palette 20
Hiding the Tool Palette 21
Status Display 21
Orientation Control 21
Schematic Drawing Tools 22
Use of the Pointer or Cursor 22
Parts Palette 24
Moving the Parts Palette 24
Hiding the Parts Palette 25
Choosing a Library 25
Selecting a Part 25
Parts Palette Pop-up Menu 26
Sheet Info Palette 26

Chapter IV - Schematic Editing 27

Design Structure 27
What is a Design? 27
What is a Circuit? 28
Types of Objects in a Circuit 28
Classes of Devices 29
Device Libraries 29
How Devices are Created and Stored 30
Device Names 30
Drawing Signals 30
Interconnecting Signals 31
Naming Signals and Busses 32
What Signal Names are Used For 32
Invisible Signal Names 32
Pin Numbering 33
Auto-Naming Features 33
Text and Picture Objects 34
Creating a Text Block 35
Editing random text 35
Sheet Sizes 35
Custom Sheet Sizes 36
Default Sheet Layout 36
General Editing Concepts 37

The Clipboard....37
Selection....38
Schematic Editing Procedures....38
Creating a New Design....38
Disposing of a Design....39
Selecting a Standard Sheet Size....39
Navigating Around a Schematic Page....39
Placing and Editing Devices....41
Signal Line Editing Procedures....43
Name Operations....45
Setting and Editing Pin Numbers....48
Selecting Circuit Objects....49
Deselecting a Selected Object....51
Creating a Text Notation....51
Creating a Picture....53

Chapter V - Advanced Schematic Editing....55

Device Packaging....55
Packaging vs. Auto-Name Assignment....55
Auto Packaging....56
Batch Packaging....57
Manual Packaging....57
Auto-Generated Name Format....58
Device Packaging Level....59
Back Annotation and Packaging....59
Packaging in Hierarchical Designs....60
Using Device Libraries Without Packaging Information..61
Packaging Attribute Fields....61
Device and Signal Auto-Naming....64
Device Auto-Naming vs. Packaging....64
Enabling Auto-Naming....64
How Names are Generated....64
Selecting an Alternate Prefix Field....65
Device and Signal Token Values....65
Multipage Schematics....66
Sheet Border Setup for Multi-Page Designs....66
Inter-page Connections....66
Bussing....69
Properties of Busses....69
Properties of Breakouts....70
Bus Pins....72
More on Text and Picture Objects....73
Background and Border Objects....73
Using Text Variables....73

More on Sheet Sizes 76
Custom Sheet Sizes 76
Power and Ground Connections 76
Power and Ground Connections in Attributes 76
Signal Connector (Power and Ground) Symbols 78
Specifying Package Type Information 80
Connectors and Discretes 80
Handling Connectors 80
Handling Discrete Components 81
Advanced Editing Procedures 82
Adding a New Page to a Circuit 82
Creating a Title Block 82
Creating a Custom Sheet Template 83
Creating Custom Sheet Border Graphics 85
Setting Device Packaging Options 86
Bus Operations 87
Setting Text Styles 90
Setting Page Reference Format 91
Setting an Alternate Name Prefix Field 92

Chapter VI - Attributes 93

Attribute Organization 93
Attribute Definition Table 93
Predefined Fields 94
User-defined Fields 94
Primary vs. Secondary Fields 94
Definition vs. Instance Fields 94
Name and InstName Fields 95
Value List Fields 97
Default Position Fields 98
Default Values 99
Attribute Field Options 99
Attribute Limitations 102
Attribute Procedures 103
Defining a New Field 103
Editing Attribute Data (General) 104
Editing Device Attribute Data 105
Displaying an Attribute on the Schematic 105
Rotating Attribute Text 106
Probing Attributes on the Schematic 106
Setting Attribute Text Style 106
Creating a Value List Field 107

Chapter VII - Hierarchical Design 109

General Concepts 109
What is Hierarchy? 109
A Simple Hierarchy Example 109
Definition vs. Instance 111
Hierarchy Mode 112
Netlists from Hierarchical Designs 113
Hierarchical Names 114
Hierarchical Block Primitive Type 114
Printing Hierarchical Designs 115
Port Interface 116
Port/Pin Naming 116
Port Pin Type 117
Bus Ports 117
Power and Ground Connections 118
Auto-Generated Internal Circuit Template 118
Hierarchy Procedures 119
Setting the Hierarchy Mode 119
Creating a Hierarchical Block - Top Down 119
Creating a Hierarchical Block - Bottom Up 121
Changing the Hierarchical Name Separator 126

Chapter VIII - Before Starting a Major Design 127

Hierarchical Design 127
Hierarchy Mode 127
Pure Hierarchy Mode Restrictions 128
Hardcopy Output Format 128
Printing 128
Plotting 128
Interfacing with PCB Layout Systems 129
Naming restrictions 129
Part Names and Package Codes 129
Interfacing with SPICE 130
SPICE Data Field 130
SPICE Name Prefix 130
Hierarchical Designs for SPICE 130
Pin Ordering for SPICE 130

Chapter IX - Schematic Menu Reference 133

File Menu 134
New Design 134
Open Design 135
Close Design 135

Save Design and Save Design As....135
Revert....135
Save Page Pict....136
Print Design....137
Page Setup....137
Print Range....137
Libraries Sub-Menu....139
New Lib....139
Open Lib....139
Close Lib....140
Edit Part....140
Lib Maint....140
Edit Menu....143
Undo....143
Redo....144
Using the Clipboard....144
Clear....146
Duplicate....147
Point....147
Text....148
Zap....148
Draw Sig....149
Draw Bus....149
Attribute Probe....149
Select All....150
Show Tool Palette/Hide Tool Palette....150
Drawing Menu....151
Go To Selection....151
Pages....151
Screen Scaling Commands....153
Magnify....154
Design Preferences....155
Import Sheet Info....158
Custom Sheet Info....158
Center in Page....161
Options Menu....162
Orientation....162
Get Info....163
New Breakout....172
Push Into....174
Pop Up....174
Attach Internal....175
Detach Internal....177
Part Type....177

Packaging....178
Define Attribute Fields....178
Set Design Attributes....180
Part Type Sub-Menu....181
Make Unique Type....181
Update from Lib....181
Save to Lib....183
Packaging Sub-Menu....184
Auto Packaging Enabled....184
Default Prefix....184
Repackage Design....185
Rescan Design....185
Free Unit Report....185
Reassign Device Names....186
Device Pop-Up Menu....187
Device Info....187
Attributes....187
Name....188
Push Into....188
Flip and Rotate Commands....188
Cut....188
Copy....188
Duplicate....189
Clear....189
Attribute Value List Sub-menus....189
Signal Pop-Up Menu....190
Signal Info....190
Attributes....190
Name....190
Pin List....191
Cut....192
Copy....192
Duplicate....192
Clear....193
Attribute Value List Sub-menus....193
Pin Pop-up Menu....194
Pin Info....194
Attributes....195
Bus Pin Info....195
Attribute Value List Sub-menus....197
Circuit Pop-up Menu....198
Normal Size/Reduce to Fit/Enlarge/Reduce....198
Pop Up....198
Circuit Info....198

Pages....198
Attribute Pop-up Menu....199
Edit....199
Justification....199
Hide....200
Clear....200
Duplicate....200
Rotate Left/Rotate Right....201
Show Field Name....201

Chapter X - Device Symbol Editing....203

Introduction....203
Installation....203
Starting the DevEditor....203
The DevEditor Window....204
Graphic Palettes....205
Part Pin Tools....206
Pin Name List....207
Shift Key Usage....208
Pin Information Window....209
Pin Information Fields & Buttons....209
Pin Name List Buttons....211
Changing the Selection....212
DevEditor Menu Commands....213
DevEditor Menu....213
File Menu....216
Edit Menu....217
DevEditor Procedures....218
Creating a New Part....218
Creating a Part With Subcircuit....220
Editing an Existing Part....225
Warnings About Editing....225
Saving the Part Again....226
Assigning Attributes....227
Adding Pins....227
Automatically Creating Symbols....229
Assigning a Behavioral Model....230
Assigning a Primitive Type....231
Creating a Gate....232
Creating a Breakout....232
Creating a Power (Signal) Connector....232
Creating a Port Connector....233
Creating a Page Connector....234
Adding Packaging Information....234

Chapter XI - Report Generation 239

Introduction to Report 239
Report Features 239
General Information on Report 239
Installation 240
Using Standard Report Formats 241
Viewing Format Notes 241
Form File Naming Convention 242
Common Changes to Standard Report Forms 242
Report Operation 244
Selecting a Form File 244
Generating a Report 245
Circuit Checking 245
Report Form Examples 247
Basic Netlist Formats 248
Basic Bills of Materials 250
Spreadsheets and Word Processors 252
Other Report Generation Examples 253
Industry Netlist Formats 256
Report Form Language Reference 265
Overview 265
DesignWorks Object Types 265
Form File Structure 267
Definition Commands 269
File Control Commands 276
Pin Format Commands 277
Format Notes Section 281
Page Header Format 282
Assigning Default Names 282
Selection and Sorting Commands 284
Listing Commands 292
Hierarchical Net Listing 298
Report Command Summary 305
Definition Commands 305
Item Format Commands 306
Page Header Format 306
Selection and Sorting Commands 307
Listing Commands 307

Chapter XII - Browser Tool 309

Introduction 309
Installation 309
Memory Usage 309

Starting the Browser 309
The Browser Window 310
Updating the Browser Window 311
Browser Procedures 312
Selecting Which Objects are Displayed 312
Determining Where to Search for Objects 312
Displaying Attributes 313
Changing Attribute Values 313
Using the Buttons 315
Adjusting the Spreadsheet 316

Chapter XIII - Find and ErrorFind Tools 317

Introduction 317
Installation 317
Memory Usage 317
Starting Find or ErrorFind 317
The Find Window 318
Search Scope 318
Search Criteria 319
Information Fields 319
Find Controls 319
The ErrorFind Window 320
Search Scope 320
Device Errors 321
Signal Errors 321
Pin Errors 321
Ignoring Errors 322
Information Fields 322
Other Controls 322

Appendix A - Device Pin Types 323

What Pin Types are Used For 323
Pin Types Table 323

Appendix B - Primitive Device Types 325

Schematic Symbol Primitive Types 325
Pseudo-Device Primitive Types 326
Simulation Primitive Types 326

Appendix C - Predefined Attribute Fields 329

Primary Fields 329
Secondary Fields 331

Appendix D - Setup File Format **335**
Library and Circuit Folder 335
Tools Folder 336
Default Libraries 336
Default Design 337
Color Settings 337
Zoom Factors 339
Reduce/Enlarge Settings 339
Normal Size Setting 339
Pin Spacing 339
Breakout Parameters 340
Internal Error Checking 340
Appendix E - Internal Error Detection **341**
Credits **343**
Technical Support **344**
Bulletin Board System 344
Index **345**

This page is intentionally left blank.

Chapter I - Introduction

DesignWorks™ 3.0 is a powerful schematic editing tool with features designed to allow it to form the core of your electronics design system.

The package covered by this manual includes a number of tools interconnected by the MEDA™ framework:

Schematic	Handles all basic schematic drawing functions.
DevEditor	Provides device symbol creation and editing functions.
LibIO	Handles storage, retrieval and maintenance of device library files.
Browser	Provides spreadsheet-like access to all text attributes in a design and eases navigation in large designs.
Find	Locates objects by name or attribute value.
ErrorFind	Checks the design for possible errors.
Report	Generates netlists and bills of materials in standard or user-definable formats.

The functions of each of these tools are described in detail in this manual. Other optional packages such as Digital Simulation and EDIF Schematic Translation are described in separate manuals.

Where to Start

We suggest you ease yourself into the world of schematic editing with DesignWorks by taking the following steps:

STEP 1	Install the package using the procedures outlined in Chapter II - Getting Started and any separate release notes supplied with the package.
STEP 2	If you are using DesignWorks for the first time, work first through the separate demonstration manual supplied with the package. It provides step-by-step instructions for basic schematic editing.
STEP 3	Refer to Chapter IV - Schematic Editing in this manual for background on basic editing techniques.
STEP 4	Before you start a major design, refer to Chapter VIII - Before Starting a Major Design. This will help you make decisions that will save time later!

Chapter Organization

Each chapter is divided into two parts:

- the first sections give background information on the features covered in that chapter.
- the last section gives specific "how to" procedures for using the features.

The DesignWorks/Schematic manual is divided into the following chapters:

Chapter II - Getting Started	Covers installation and initial startup.
Chapter III - User Interface	Provides detailed information on the use of windows, menus, keyboard commands and other user controls and displays.
Chapter IV - Schematic Editing	Discusses the structure of a circuit diagram and how to create one from scratch.
Chapter V - Advanced Schematic Editing	Provides information on advanced features such as: device packaging, power and ground connections, busses, text variables, etc.
Chapter VI - Attributes	Discusses the text attribute features of the package in detail.
Chapter VII - Hierarchical Design	Covers the hierarchical design features of DesignWorks.
Chapter VIII - Before Starting a Major Design	Discusses a number of considerations that may affect how you organize a design.
Chapter IX- Schematic Menu Reference	Describes menu commands in detail.
Chapter X - Device Symbol Editing	Provides information on creating and editing device symbols.
Chapter XI - Report Generation	Reviews the use of standard netlist formats and provides details on creating custom formats.
Chapter XII - Browser Tool	Describes the Browser attribute editing and schematic navigation tool.
Chapter XIII - Find and ErrorFind Tools	Describe using the Find and ErrorFind tools to locate devices and signals on your schematic.

New Schematic Features in 3.0

Hierarchy

DesignWorks 3.0 implements a true hierarchical design structure. This means that "device" symbols appearing in a circuit can actually represent another, nested circuit. This sub-circuit can be opened and edited at any time in separate circuit window.

DesignWorks 3.0 has three hierarchy modes to address different design situations:

- "Flat" Mode - This is the simplest mode and is intended for small designs or those imported from older versions of DesignWorks.
- "Physical" Mode - This mode allows full use of hierarchical design, while still allowing data associated with individual device instances to be stored at any point in a design. It is intended for larger board-level or FPGA designs where interactive simulation is used or PCB layout information is to be kept with the design.
- "Pure" Mode - This mode stores only a single set of internal data for each type of sub-circuit in a design. This allows very large designs to be represented in a small amount of memory, but does not allow interactive simulation or physical layout data to be associated with the design. It is intended for VLSI or other large-scale design applications.

See Chapter VII - Hierarchical Design for more information.

Attributes

Each design has associated with it an "attribute definition table" which lists all the fields which will be used in the design. Fields can be associated with any device, signal or pin or with the design itself. More than 50 fields are predefined for a variety of applications, and you can add more at any time. Once a field name is defined, its name is chosen from a list of available fields when entering data. This greatly reduces that chance of entry error.

Other new attribute features include:

- Attributes can now be displayed at any position on the diagram, independently of the name.
- The same attribute data can be displayed in multiple locations.
- All text displayed on the diagram can be edited right in place.

- Attribute text can be rotated in 90° increments.
- Each field can contain up to 32,000 characters of data per object.
- Any text characters and even binary data is valid in an attribute field.

See Chapter VI - Attributes for more information.

Sheet Templates

DesignWorks 3.0 allows all aspects of sheet sizes, borders, grid layouts, title blocks and text annotations to be customized. The package now has no built-in knowledge of standard sheet sizes, but instead imports all information from "template files", which are actually just empty design files. Text and graphics can be placed on a sheet and marked as "border" items. They will then be automatically copied to each new page in the design and can be imported to other designs with a single command.

See Chapter V - Advanced Schematic Editing for more information.

Packaging

DesignWorks 3.0 has the ability to automatically assign gates to packages, either interactively or on demand. A package table associated with the design keeps track of all package names that have been used and what units are available in each.

This feature also has the effect of automatically naming all devices, even those that do not have multiple gates per package. Names are generated using a prefix field which can be stored in the library or can refer to a default prefix for the design.

See Chapter V - Advanced Schematic Editing for more information.

Browser Tool

The new Browser tool allows you to view and edit attribute data for an entire design in a spreadsheet-style format. This saves time locating and selecting devices or signals across multiple pages and hierarchy levels.

See Chapter XII - Browser Tool for more information.

Other Features

• The DevEditor tool allows any circuit to be assigned to a symbol. Symbols can be created using the built-in drawing tools or imported from any external drawing program.

• Thousands of standard device symbols are provided, including the 74XX series, 4000 series, ECL, microprocessors, memories, discretes, etc.

• A circuit schematic can consist of up to 1000 pages, each one being a single or multiple standard sheets, up to a total of 5 feet by 5 feet. Each circuit page is displayed in a separate window each with independent control of scroll and zoom. Logical connections are allowed between pages using a page connector symbol. Hierarchy levels can be nested to any depth.

• Any number of circuit windows can be open simultaneously, allowing easy copying of partial or complete diagrams from one file to another.

• Commands and drawing modes can be selected using menu items, keyboard equivalents or a floating tool palette.

• Any group of objects on the drawing can be repositioned with a simple click-drag mouse action. Signal lines are rerouted interactively to maintain right angle connections.

• Multiple signal line routing methods allow most pin-to-pin connections to be made with only two mouse clicks.

• Any combination of named signals can be represented by a single bus line and any subset of these can be brought out through a breakout at any point.

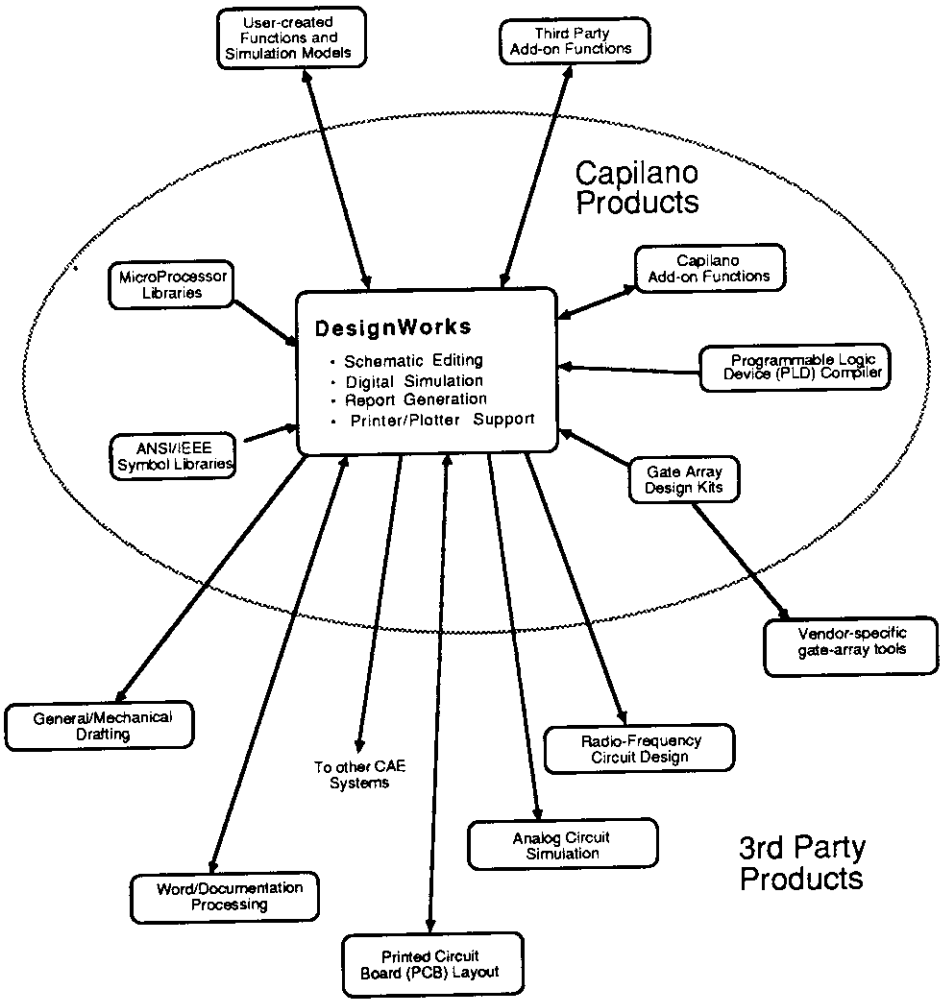
• Signal names are global across a single page and across multiple pages when a page connector symbol is added to the signal line. Like-named signal traces on a single page are thus logically connected for simulation and netlisting purposes.

• Drawing borders and reference grids can be drawn automatically. Arbitrary pictures and text created in other programs can be pasted onto a circuit schematic. Similarly, complete or partial circuit diagrams can be pasted into word processing or drafting documents.

• Objects can be drawn in user-selectable colors on machines equipped with a color display.

• Circuit diagrams can be printed on the ImageWriter™ or LaserWriter™, or written as PICT (MacDraw™) files for incorporation into other documentation. A plotter utility package is included which will plot circuit files on Houston Instruments, Hewlett-Packard and Roland plotters.

DesignWorks Interfaces to the CAE World



Interface Features

• The report generator tool produces custom-formatted net and component lists from DesignWorks circuit files. These output files can incorporate user-defined device and signal attributes such as special part numbers, manufacturer or supplier names, power consumption, package type, cost, or any other information. This allows complete production circuit data to be

passed to other equipment or software, or included in documentation. Standard form files produce output for numerous industry netlist formats.

• Back-annotation to DesignWorks schematics is available for many popular PCB layout and simulation systems. This is done using add-on tools specific to the target system and these are sold separately.

• The powerful MEDA™ module interface allows specialized functions to be added to DesignWorks without modification of the program itself.

Limitations In This Version

• Maximum number of pages in a circuit is 1000.

• Absolute maximum number of devices in each circuit level is 32767. There is no limitation on circuit nesting depth.

• Typical number of devices without severe performance degradation is 500 - 1000 on a Macintosh Classic, 8000 - 12,000 on a Macintosh Quadra.

• Maximum length of a device, signal or attribute field name is 16 characters.

• Maximum length of a library part name is 32 characters.

• Maximum length of a pin number is 4 characters.

• Maximum number of characters in an attribute block is 32000.

• Maximum number of pins on a device is 800.

• The entire design must fit into available memory.

Compatibility

• Compatible with all current Macintosh models with 4 megabytes of memory or more, including the Macintosh Classic™-series, II™-series and Quadra™-series.

• Requires Macintosh System Software version 6.0.5 or later and is fully System 7.0 compatible.

• Works with all Apple and most third-party Chooser-selectable output devices, including LaserWriter II™ and ImageWriter™ printers.

Introduction to MEDA™

The MEDA (Modular Electronic Design Application) concept arises from the desire to create an environment in which independently-developed electronic design functions can operate in an integrated and interactive fashion. This benefits users by allowing them to work in a consistent environment and giving them painless access to a variety of special-purpose packages. In addition, it allows Capilano Computing, independent software developers, and users with programming capability to add specialized, yet fully integrated, functions to DesignWorks.

Example application modules that are well suited to this concept are:

- new simulation models for the existing simulator
- a complete, new simulation module, such as analog circuit simulation
- interfaces to external simulation hardware
- various types of circuit analysis, such as timing analysis and fault analysis
- back annotation
- logic synthesis, generating netlist or PLD code output
- integrated PCB or chip layout
- schematic generation from netlists
- control of external instrumentation, such as logic analyzers, digital oscilloscopes, pattern generators, etc.
- novel ways of displaying simulation data

The tools provided as part of the DesignWorks package all communicate using the MEDA interface.

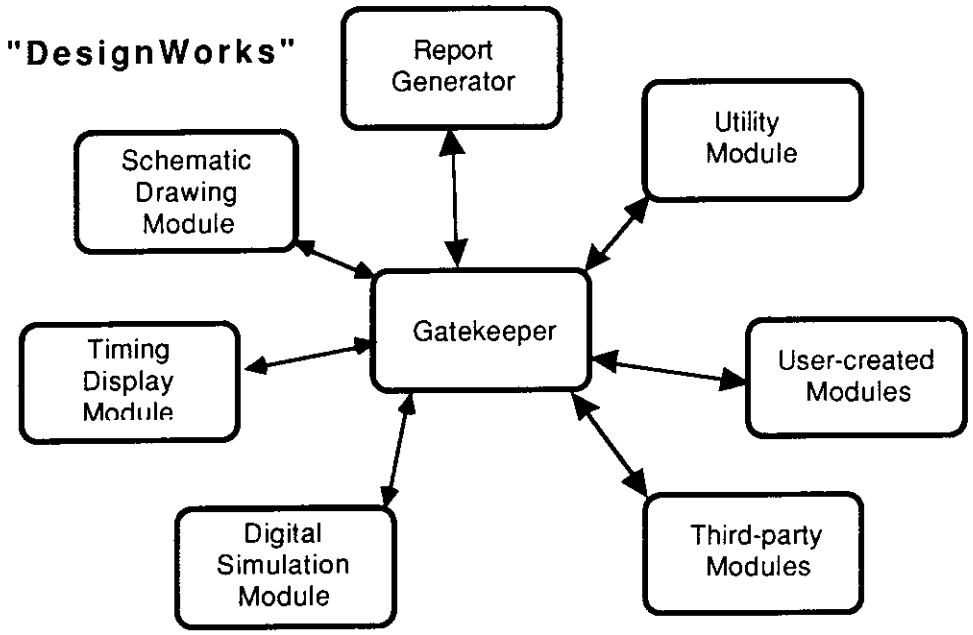
The MEDA Concept

In the MEDA concept, a central application, referred to as the "GateKeeper" makes available general services, such as:

- time sharing
- general circuit drawing facilities
- window management
- menu management
- utility I/O, such as an editable text window
- debugging facilities, such as a message window

In addition to the services provided by the GateKeeper, client modules can be simplified given that related functions can be provided by other clients. For example, all the schematic entry and simulation functions implemented in DesignWorks are available through the MEDA interface, including the ability to query or modify circuit connectivity, attributes, simulation information, etc. This means that modules requiring circuit editing or simulation functions can make use of the existing ones rather than having to re-implement them.

MEDA Inter-Module Communications



Client modules communicate with the GateKeeper or with each other via a message-passing scheme. A simple time-sharing task manager is implemented allowing the modules to coexist under the current single-task Macintosh operating system, but the communications scheme is designed with multi-tasking and multi-processing in mind.

As the above diagram shows, the major functional parts of DesignWorks are internally implemented as MEDA modules, even though they are combined into a single file in this release. Other functions, such as the Report Generator, are implemented in separate files. In either case the communication method is the same.

Creating MEDA Modules

MEDA modules are written using the standard C programming language. A MEDA Developer's Kit package containing interface libraries, example files and documentation is available as a separate package from Capilano Computing.

Notes Regarding Copyright

The DesignWorks software and manual are copyrighted products. The software license you have purchased entitles you to use the software on a single machine, with copies being made only for backup purposes. Any unauthorized copying of the program or documentation is subject to prosecution.

Note regarding trademarks

A number of product trademarks are referred to in this manual. Full credit for these is given in the last chapter.

Chapter II - Getting Started

This section gives you information on installing and starting up DesignWorks.

> **NOTE**: Since the DesignWorks package and the Macintosh system are constantly being upgraded, there may be recent changes and additional information supplied in Release Notes with the package. It is important that you review these prior to installation as they may contain information that supersedes that given here.

We will be assuming that you are already familiar with general Macintosh operation, such as copying files, creating folders, etc. If you are not, then work through the introductory sections of the Macintosh user's guide.

IMPORTANT: Make a backup copy

Before proceeding, make a backup copy and a working copy of your DesignWorks disks, then put the originals away for safekeeping.

Installation

For more detailed information on the files included with the system, see the release notes supplied with this manual.

General Rules

In general you can place any of the files supplied in any folder or on any disk as suits your preference. The only absolute rule is that the DesignWorks Setup" file must not be renamed and must be in the same folder as DesignWorks itself, or it will not be found at startup and no initial setup will be done. The other general warning is that if you change the name or location of the Basic Libs or Tools folders, it will be necessary to make corresponding changes to the "DesignWorks Setup" file. See below for more information on how to do this.

The device libraries take up quite a bit of disk storage (around 30 megabytes for a complete installation), so if you're short of space you might want to peruse the Device Library Index (provided in paper and file form) to determine which ones you will be actively using. To really optimize space utilization, you can use the Lib Maint command functions to select only the

individual devices you need from the libraries provided and make your own set.

See Chapter IX - Schematic Menu Reference in this manual for information on how to do this.

External Tools

DesignWorks 3.0 consists of a small central application (called the GateKeeper) and a collection of external program modules which implement most of the program functions. In order for DesignWorks to recognize and load these modules, they must be in the same folder as the DesignWorks program or in the folder specified in the TOOLFOLDER setup file item. All such tools will be loaded when DesignWorks is first started and will appear in the Tools menu. Normally, the Schematic tool will be activated at startup and will display an untitled document window. Other tools with more specific functions (such as Report or DevEditor) are invoked by selecting their name in the Tools menu.

In this version, the only tools that are essential to normal operation of the program are Schematic and LibIO. In order to conserve memory, other tools can be kept aside in a storage folder except when actually needed.

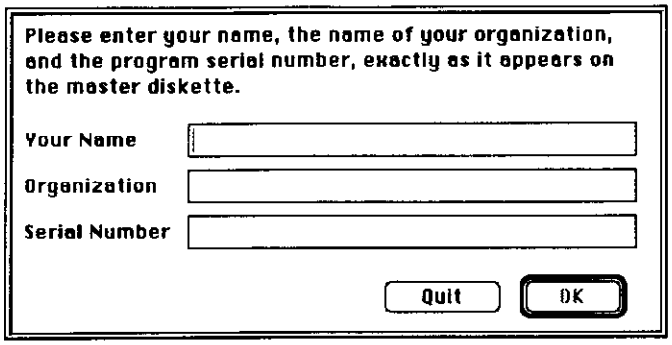
See the installation notes supplied with this package for up to date information on tool installation.

Default Library Folder

The DesignWorks Setup file allows you to name one or more folders as default library folders. All libraries in these folders will be opened automatically when the program is started. In the standard setup file provided with DesignWorks, the default folder is called "Basic Libs". See the description of the LIBRARYFOLDER keyword in Appendix D - Setup File Format.

Serial Number Entry

The first time you start the program, you will see the following box:


Please enter your name, the name of your organization, and the program serial number, exactly as it appears on the master diskette.
Your Name
Organization
Serial Number
Quit
OK

Enter the following information:

- Your name and the name of your organization. This information will be displayed each time the program is started.
- The serial number as printed on your master diskette. This is a sequence of 20 letters and numbers. It can be entered either with or without separating blanks.

After this information is entered once, it is kept with the program and you will not need to enter it again.

DesignWorks uses a network copy protection scheme that limits the number of users of a single copy on a network. The serial number contains encoded in it the number of authorized users and validation codes for add-on tools. You can upgrade your copy at any time by contacting Capilano Computing at the phone number given elsewhere in this manual.

Correcting or Re-entering the Serial Number

If for any reason you should wish to re-enter this information, you can hold the Option key pressed while starting the program from the Finder. The program will then redisplay the serial number entry box.

Troubleshooting Installation

PROBLEM: Program seems to run, but no windows appear and the Window and Tools menus have nothing in them.

CAUSE: The program has failed to locate any of the tool files. This could be due to any of the following:

- The tools are not in the same folder as the DesignWorks™ 3.0 application or the Tools folder has been renamed.

- The DesignWorks Setup file has been renamed or is not in the same folder as the DesignWorks™ 3.0 application.
- The folder name specified in the TOOLFOLDER line of the DesignWorks Setup file is incorrect.

PROBLEM: Error message about "Could not open library XXXX"

CAUSE: The name or location of a library specified in the DesignWorks Setup file is incorrect.

PROBLEM: Error message about "Unexpected changes have been made to the application..." appears on startup.

CAUSE: The program checks itself for unexpected changes each time it starts up, as a precaution against viruses. If you see this message, you should run a virus checking program on your disk and re-install the program from the master diskettes (being careful to keep them write-protected). In most cases, you can continue and execute the program without major ill effects, but there is a risk that you will contaminate or corrupt other files in your system.

PROBLEM: Error message about "Unable to allocate memory or bad format for module XXXX".

CAUSE: This message can appear for one of the following reasons:

- The module in question is obsolete, i.e. it is part of an older DesignWorks release.
- The module file is corrupted.
- There was insufficient memory to load the module.

File Compatibility

DesignWorks 3.0 uses a completely new file format which allows for the new design structure and provides improved flexibility for future expansion. For this reason, the new version will not directly read files created by earlier versions, nor can it write files compatible with earlier versions. However, files created by all earlier versions of DesignWorks™, LogicWorks™, LogiMac™ and Douglas Schematic can be converted to the new format using the Converter tool.

Converting Old Files

For complete information on conversion, see the separate document "Converting to DesignWorks 3.0 from older versions".

The Converter tool is invoked by selecting its name in the Tools menu. It reads in the old file, performs some internal conversions and then writes out a file in the new format. The new file can then be Opened in the normal way in the Schematic tool.

Memory Usage

If you are running DesignWorks 3.0 under MultiFinder in Apple's System 6.0.x or under System 7.0.x, we recommend a memory size setting of at least 3000K (more if you frequently work with large design files). This number can be set by clicking once on the DesignWorks™ 3.0 program icon while in the Finder, and doing a Get Info (⌘-I) menu command.

In the current release a number of new features have been added which significantly increase the amount of memory occupied by the program itself, although the memory occupied by the circuit data has actually been reduced somewhat. MEDA modules are loaded when the program is run and occupy memory whether they are being used or not.

The Setup File

The setup file is a text file which specifies some initial actions DesignWorks will take each time it is started up. The setup file specifies the following information:

- which libraries to open when the program is started.
- which colors to use in displaying objects on the Macintosh II screen.
- other options.

The setup file can be created or edited using your favorite programming editor or word processor. If you use a word processor, be sure to save the file using a "text only" option. The setup file must be called "DesignWorks Setup" and must be in the same folder as the DesignWorks program itself.

A typical setup file looks like the following:

```
LIBRARYFOLDER :Basic Libs:;
TOOLFOLDER :Tools:;
```

The first word on each statement is a keyword which specifies a setup option. Statements are terminated by a semi-colon and can contain embedded comments in braces. The most commonly-used setup items are:

- LIBRARYFOLDER specifies the folder to look in for device library files to open.

• TOOLFOLDER specifies the folder to look in for code module files.

For more information on the meaning and format of these statements, see Setup File Format in the Appendix.

Chapter III - User Interface

This chapter provides general information on the use of windows, drawing tools and other user-interface features of DesignWorks.

Mouse Button Usage

Three different mouse button actions are used for various functions in DesignWorks. For clarity we will use the following terminology when referring to these actions in the remainder of the manual:

Click - means press and release the button without moving the mouse. Example: To select a device, click on it.

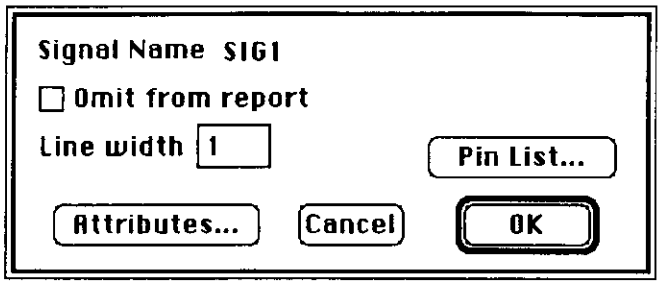
Click and drag - means press the mouse button and hold it pressed while moving the mouse to the appropriate position for the next action. Example: To move a device click and drag it to the desired new position.

Double-click - press and release the button twice in quick succession without moving the mouse. Example: To open a device's internal circuit, double-click on the device.

Dialog Boxes

Many DesignWorks functions require information to be displayed or prompted from the user. To do this a special window called a "Dialog Box" is displayed, such as the following one which is used when a Get Info command is executed for a signal.

In general, the controls in dialog boxes will behave according to Macintosh standards, with the following exception: Since many of the boxes require text entry with multiple lines, the Return key does not normally cause the box to be closed and the default action to be performed; this function has been given to the Enter key. If the flashing text cursor is currently in a text box that allows multiple line entry, such as the attributes box, then the Return key will simply insert a carriage return in the text. If the insertion point is located in a text box that does not allow multiple lines, then the Return key will behave the same as the Tab key and cause the next text box to be activated.



In most dialog boxes requiring text entry, the keyboard equivalents for the clipboard command Cut (⌘-X), Copy (⌘-C) and Paste (⌘-V) are active and can be used to transfer text to or from a text box.

Window Usage

Circuit Windows

Each circuit window displays one page of a circuit schematic. The title on a circuit window will be the name of the circuit file with the number of the page displayed in that window, as in "Interface Board:3". Any number of pages in a given circuit and any number of circuits can be displayed simultaneously. At any given time only one page of one circuit is "current", that being the one in the topmost window. Any other window can be made current simply by clicking the mouse anywhere in the window.

Tool Windows

External MEDA modules, or Tools, can create their own windows which may be displayed concurrently with other DesignWorks windows. In general these obey the normal rules for resizing, etc., but their function is totally defined by the module.

Closing a window

Clicking in the Go-Away box on a Circuit window has the effect of closing the circuit page. If the page being closed was the only one open on the circuit then the circuit is closed.

Redisplaying a window

When a circuit page window is closed it is removed from the Window menu to avoid clutter. To reopen a page in the current circuit level, use the Page command in the Drawing menu. To reopen an internal-circuit window, double-click on the device in question.

Resizing a window

To enlarge or reduce a Circuit window, position the pointer in the size box at the lower right corner of the window and press and hold the mouse button. As long as the button is held down, a gray outline of the window will track the mouse movements. When the button is released the window will be redrawn with the new size and shape.

To expand the circuit window to fill the whole screen, click in the "Zoom" box at the upper right corner of the window. Clicking again in this box will cause the window to return to its original size.

Moving a window

To move the Circuit window, position the pointer in the title bar anywhere along the top edge of the window and press and hold the mouse button. As long as the button is held down, a gray outline of the window will track the mouse movements. When the button is released the window will be redrawn at its new position.

The Window Menu

The Window menu provides a means of bringing to the front any window currently open.

Keyboard Usage

The Macintosh keyboard is only absolutely required when entering names for devices or signals, or for placing random text notations on the drawing. However, the command (⌘), shift and option keys on the keyboard can be used with many editing operations to invoke special features such as auto-alignment, auto-naming and different signal line drawing methods. In addition the arrow "cursor" keys (if available) can be used as a convenient

way of setting symbol rotation while placing devices or pasting circuit groups. These options are described in detail in the relevant chapters.

Pop-up Menus

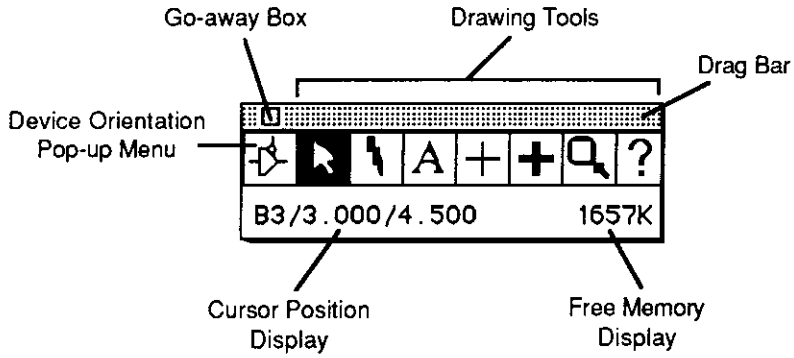
At any time while editing a diagram you can hold the ⌘ key and click on a schematic object. A pop-up menu will appear under the cursor allowing you to select from commands appropriate to that object. E.g. the menu for a device contains commands to get device information, edit attributes, open the internal circuit, flip or rotate the symbol, Cut and Copy operations, change gate package assignment, etc.

Separate pop-up menus are available for devices, signals, pins, attributes, and (if clicking in open space on the drawing) the circuit itself.

For details on commands in pop-up menus see Chapter IX - Schematic Menu Reference.

Tool Palette and Status Display

The tool palette is a small window which always remains in front of any circuit windows currently open. The individual items in this display are described in their own sections below.



Moving the Tool Palette

The tool palette can be moved to any desired position on the screen by clicking and dragging in its drag bar.

Hiding the Tool Palette

The tool palette can be removed from the screen by clicking once in its "go-away" box. To re-display the palette, select the Show Tool Palette menu item in the Edit menu.

Status Display

The lower area of the tool palette window is used to optionally display several items of status information.

Cursor Position Display

The cursor position is shown as a grid reference and an X and Y distance in inches from the sheet origin (which may be any corner, depending on sheet setup). The cursor position display can be enabled or disabled in the Design Preferences command in the Drawing menu.

Free Memory Display

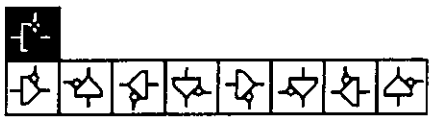
If enabled, the amount of free memory available for circuit data is displayed in the lower right corner of the tool palette window. This number is shown in Kbytes, i.e. multiples of 1024 bytes. The free memory display can be enabled or disabled in the Design Preferences command in the Drawing menu.

Orientation Control

The orientation icon shows the orientation that will be used when the next device or circuit scrap is pasted into the schematic. The orientation can be changed by clicking on the icon, using the arrow keys on the keyboard, or by using the Orientation menu command.

Orientation Pop-up Menu

Clicking and holding the orientation icon in the tool palette will cause the following pop-up menu to appear:



Choose any one of the 8 possible orientations by sliding across this menu. The selected orientation will be displayed in the tool palette icon.

This selection has no effect on items already placed in the schematic. It affects only the flickering image of items currently being placed and future devices selected from the parts palette.

More information on symbol rotation is given in Chapter V - Advanced Schematic Editing.

Schematic Drawing Tools

The seven drawing tools are used to set the cursor mode for schematic editing. More information on cursor modes is given in the following section.

Use of the Pointer or Cursor

In subsequent descriptions, we will refer to the small shape which tracks the mouse position on the screen as the "pointer" or "cursor". In DesignWorks there are a number of different cursor modes used which determine what action will be performed when the mouse button is clicked. Following is a summary of the cursor modes. More detailed descriptions of operations performed in each mode are provided in later reference chapters.

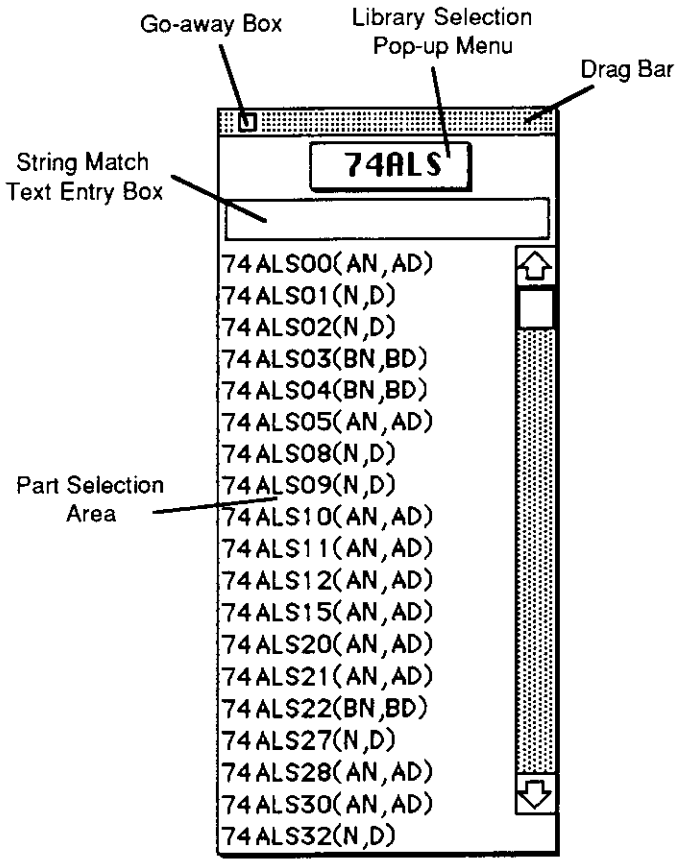
Note that the cursor shape sometimes differs from the tool palette icon for ease of pointing.

Tool Palette Icon	Initial Cursor Shape	Equivalent Menu Command	Description
		Point	Used to select or drag objects, extend signals.
		Draw Bus	Used to create a new bus line or extend an existing bus. Clicking once fixes a corner, double-clicking terminates the line.

+	×	Draw Sig	Used while creating a new signal line or extending an existing signal. Clicking once fixes a corner, double-clicking terminates the line. Note that most signal drawing operations can also be done in Point mode.
A		Text	Used to select a signal or device to name, or to place random text on the diagram. Point at the item you want to name and press and hold the mouse button. Move to where you want the name to appear, then release the button.
		Zap	Used to remove single objects. Press the button to remove whatever the tip of the cursor is pointing at. Objects can also be removed in groups by selecting them and using the Clear command or delete key.
		Magnify	Used to zoom in and out. Clicking on a point or dragging down and right zooms in, dragging up and left zooms out.
?	?	Attribute Probe	When the "dot" portion of the cursor is clicked on a device, pin or signal, the contents of the primary attribute fields for that object are displayed.

Parts Palette

Parts library contents are displayed in a floating palette window which looks like the following:



The Parts palette displays the contents of the selected library file and allows any item to be selected for placement in the schematic.

Moving the Parts Palette

The Parts palette can be moved to any desired location on the screen by clicking and dragging in its drag bar.

Hiding the Parts Palette

The Parts palette can be removed from the screen by clicking in its "go-away" box. To re-display the palette, select the Show Parts Palette command in the Libraries sub-menu of the File menu.

Choosing a Library

The palette displays only the contents of only one library at a time. The library currently being displayed is shown in the library selection box at the top of the palette. Clicking in this box allows you to select from among any of the open library files. Library files are opened by selecting the Open Lib command in either the Libraries sub-menu of the File menu, or in the Parts palette pop-up menu discussed below.

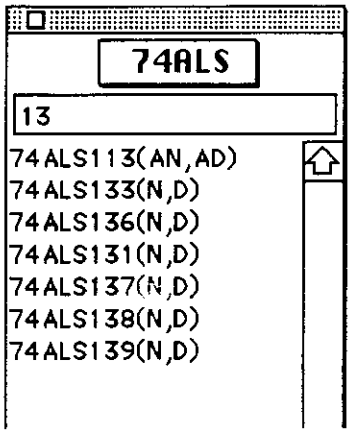
Selecting a Part

To select a part for placement in the schematic:

1) If necessary, use the scroll bar to scroll the list until the desired part name is in view.
2) Double-click on the part name in the list.
3) Move the cursor to the current schematic window.

Using the String-Matching Box

The string match text box allows you to type characters which will reduce the size of the list and make it easier to locate the desired part. Simply click in the text box and type the desired characters. After a brief pause, the displayed part list will be reduced to only those parts that contain the typed string of characters. For example:



To return to the full selection, click at the end of the characters in the box and backspace over them until the box is cleared.

Parts Palette Pop-up Menu

As a short-cut, the Libraries sub-menu in the File menu can be popped up right in the Parts palette. Hold the ⌘ key on the keyboard while clicking and holding anywhere in the Parts palette window. Menu selections made here are equivalent to selecting the same item in the Libraries sub-menu.

See Chapter IX - Schematic Menu Reference for specific information on these items.

Sheet Info Palette

Custom sheet sizes and borders are created using the palette displayed using the Custom Sheet Info command in the Drawing menu.

This specialized operation is discussed in Chapter V - Advanced Schematic Editing.

Chapter IV - Schematic Editing

This chapter provides information on the elements of a DesignWorks design and how one is created.

Design Structure

What is a Design?

In DesignWorks the term "design" refers to a complete, independent logical entity. The following rules outline how a design is stored:

- A single design is stored in a single file and no logical connections are made between designs. All information required to display and edit a design is stored in the design file.
- A design never makes reference to external library files. When a library device is used, all information needed is read from the library and stored with the design. Changing the original library definition will not automatically update the design. Library devices can be updated manually using the Update from Lib command.
- A design has a top-level circuit, referred to as the *master* circuit. This circuit may contain any number of devices which themselves can contain circuits, called *sub-circuits*. Sub-circuits can be nested to any desired depth, limited only by available memory.
- When a design file is opened, the entire contents of the design are read into memory. This means that design sizes are limited by the available memory in your computer and increasing the memory allocated to the program will increase the size of the designs you can work with.
- A number of user-settable parameters are stored with the design and affect the entire design when changed. These include:
  - attribute and pin number text style settings
  - the attribute definition table
  - signal and device auto-naming settings
  - display options, such as crosshairs and printed page breaks
  - page reference format
  - hierarchy mode

What is a Circuit?

A design consists of a single master circuit with any number of nested sub-circuits.

- Each master circuit or sub-circuit can consist of one or more pages (up to 1000 in this version). The user can elect to draw the entire circuit on a single page or divide it up functionally onto a number of pages.
- Each page is viewed in a separate circuit window and any or all pages in a circuit can be displayed on the screen simultaneously.
- A page is drawn on the screen as if it was a single piece of paper, although it may have to be broken up into a number of individual sheets of paper for printing or plotting.
- Logical connections can be made between pages using the Page Connector device allowing for full simulation and netlisting across pages.
- If the circuit is a sub-circuit, then logical connections to the parent device symbol are made using the Port Connector device. Port connectors in the sub-circuit are matched by name with pins on the parent device.

Types of Objects in a Circuit

A DesignWorks circuit is made up of four types of entities: devices, signals, text objects and picture objects.

- A device is an object having a symbol, signal connection points called "pins", and optional attributes, internal circuit and simulation information. A device in DesignWorks can correspond to a physical device in a circuit, or it can represent a sub-circuit block or a *pseudo-device*, such as a page connector or bus breakout.
- A pin is a connection point on a device. A pin is not an independent entity since it only exists as part of a device and cannot be created or deleted separately. However, pins can have attributes, pin numbers and other parameters that may be different from pin to pin on the same device. The Get Info command can be used on a selected pin to view and set pin parameters. A *bus pin* is a special type of pin that represents an arbitrary number of internal pins. The internal pins are not visible on the schematic but can still have the same logical properties as other pins.
- A signal is a conductive path between devices. Signal connections can be made visually by drawing lines between device pins, or logically by name or bus connection.

• A text object is used to place a title block or other notation on the diagram. Text can be typed and edited directly within DesignWorks, or can be created externally and pasted onto the diagram from the clipboard. Text objects are not associated with any other object and are not accessible through net or component lists. The attribute facilities should be used to associate text with specific devices or signals. Text operations are described later in this chapter.

• A picture object is used to place any graphics item imported from another program or the DevEditor tool via the clipboard. The Schematic tool cannot edit picture objects directly, other than moving, deleting and copying them whole. They can, however, be pasted into a DevEditor window, edited using the device symbol tools, then copied and pasted back into the schematic. Picture operations are described later in this chapter.

Classes of Devices

For the purposes of this section, device types in DesignWorks can be divided into four groups:

• Schematic symbols: These are the normal symbols used to represent physical devices on a schematic. Their simulation function, if any, is determined solely by defining an internal circuit.

• Pseudo-device types: These are the symbols used for page connectors, power and ground symbols, etc.

• Simulation types: These are special device symbols intended for use with the DesignWorks Digital Simulator option.

• "Behavioral model" devices have a simulation function defined by program code using the MEDA interface.

See a description of primitive types in DesignWorks in Appendix B - Primitive Device Types.

Device Libraries

The symbols and related parameters for DesignWorks devices are stored in data files called device libraries. Libraries can be opened and closed using the Open Lib and Close Lib commands, or using entries in the setup file.

For each device type in a library the following data is stored:

- general information on the type, such as number of pins, number inputs, number of outputs, type name, default delay, default attributes, position, orientation and type of each pin, etc.
- a picture representing the symbol for this type.
- a polygon outlining the symbol, used for highlighting and erasing the symbol.
- an optional internal circuit definition.

How Devices are Created and Stored

Libraries are created and modified using the DevEditor tool, which is described elsewhere in this manual.

See Chapter X - Device Symbol Editing for more information.

Device Names

Note re terminology: In this manual, we use the term "device name" to refer to the character string that identifies a unique device in the circuit. Typical device names might be "U23", "C4", "IC12A", etc. This is distinct from the "type name" or "part name" that is used to distinguish the type definition that is read from a device library. Typical part names are "74LS138", "MC68000L8", "SPDT Switch", etc.

Device names may contain any letters, numbers or special characters that you can type on the keyboard, but are restricted in length to at most 15 characters. The name associated with an object can be placed anywhere on the diagram and will be removed if the object is removed.

NOTE: Although DesignWorks does not enforce restrictions, we recommend not using blanks or special characters in device or signal names. This can result in problems exporting data to external systems such as PCB or simulation tools.

Drawing Signals

Signal lines are drawn in either Point mode or Signal Drawing mode.

Interconnecting Signals

If you draw a signal line such that the end of the line contacts a second signal line, then those two signals will be interconnected. Also, if you place a new device such that one of its pins contacts an existing signal line, that pin will be connected to the signal. If both of the two signals being connected were named, then you will be prompted to choose the name of the resulting signal. Whenever three or more line segments belonging to the same signal meet at a given point, an intersection dot will be placed at that point automatically.

NOTE: For efficiency, signals are only checked for connections at their endpoints and only signals actively being edited are checked. It is possible to create overlapping lines that do not connect by unusual combinations of editing operations. This situation is usually visually apparent at the time the editing is done since the intersection dot will be missing and the entire signal will not highlight when clicked on. You may wish to also use the ErrorFind tool to locate these situations.

See more information on connection checking under the Paste command in Chapter IX - Menu Reference.

Connecting Signals by Name

Signal names can be used to make logical connections between wires that are not visually connected on the schematic. The following rules apply:

- Signal names must be visible to be checked for connections, unless a Signal Connector device (e.g. Ground) is attached. More information on invisible names is given in the following section.
- Signal names are known throughout a single page. Like-named signal traces on a single page are thus logically connected for simulation and netlisting purposes. Whenever a signal name is added or changed, the circuit is checked for a change in connectivity. If the name is now the same as another signal on this page, the two signals are merged into one. If this signal segment was previously connected by name to others and the name is changed, then the logical connection is broken. Whenever a name change causes two signals to be connected, both parts will flash on the screen to confirm the connection.
- Signal names are global across all pages of the circuit when a page connector symbol is added to the signal line. Thus if the name is changed on a signal line having a page connector then all circuit pages

are checked for like-named signals having page connectors. If any such signals are found then a logical connection is made between them.

• Signals which are contained in busses are a special case. All signals contained in busses have a name, even if this is not displayed on the diagram. However, the names of bussed signals will not be used to make logical connections unless an explicit name label has been added to the signal line.

For example, if you have a bus containing a signal named CLK and a separate signal line also named CLK, there will be no logical connection between these two signals. The name appearing on the bus breakout is part of the breakout symbol and is not considered to be a name label. If an explicit label is added to the bussed CLK signal (using the Name cursor) then the two CLKs will be logically connected.

• The same rules discussed above for signals also apply to busses. Whenever two busses are logically connected, all like-named internal signals also become logically connected. Note that connecting busses across pages requires a "Bus Page Connector" device rather than a "Page Connector" to avoid compatibility problems between bus and signal connections.

Naming Signals and Busses

Names may contain any letters, numbers or special characters that you can type on the keyboard, but are restricted in length to at most 15 characters. The name associated with an object can be placed anywhere on the diagram and will be removed if the object is removed.

What Signal Names are Used For

The signal name is referenced by the following DesignWorks functions:

• The signal name is used in Report Generator output, such as netlists and simulation data reports.

• The signal can be located by name using the Find tool.

• Signals can be logically interconnected by name.

Invisible Signal Names

DesignWorks allows you to assign and edit signal names in a circuit without making them visible on the diagram. This can be used to assign

names that cannot be conveniently placed on the diagram, or that are needed for report generation purposes only. Signals with invisible names are not connected by name, except for invisible names created by a Signal Connector device, as described above.

Pin Numbering

Pin numbers may contain 1 to 4 characters and are always positioned adjacent to the associated pin.

Uses of Pin Numbers

Pin numbers are used only for labeling purposes and have no particular connectivity significance to DesignWorks. Pin numbers are not checked for duplicates or other invalid usage. Pin numbers placed on a diagram will be used in creating a netlist (see the Report Generator manual), and will appear when the circuit is printed. If a pin is unnumbered, it will appear in a netlist with a "?" unless the AUTONUMBER option is used in the report form file.

Default Pin Numbers

A device type may have default pin numbers associated with it which will appear when the device is first created. These pin numbers are not permanent and can be edited or removed by the techniques discussed in this section. Devices in a library can have default pin numbers assigned to them using the DevEditor tool.

See Chapter X - Device Symbol Editing.

Auto-Naming Features

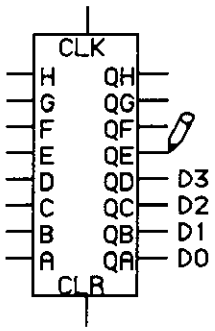
Two features are available to simplify the naming of groups of related signals, devices and pins. These features are activated by holding down the Command, Option or Shift keys while selecting the signal to be named with the Text cursor.

Auto-alignment - If the ⌘ key is held down while the signal is selected, the text insertion point will be positioned horizontally aligned with the last signal name that was entered. The vertical position is determined by the vertical position of the line that was clicked on. This feature works only with signal names, not with devices or pin numbers.

Auto name generation - If the Option key is held down while a signal, device or pin is selected, a new name is generated automatically for this item. The new name will be the same as the last one entered, except that the numeric part of the name will have been incremented. If the previously-entered name did not have a numeric part, then a "1" digit will be appended to it. If the Shift key is pressed at the same time, the number will be decremented instead of incremented.

Sequential Naming - The above two features can be used in combination to perform easy naming of sequential signals. The normal symbol standard in DesignWorks is to position the highest numbers at the top, so you can either:

- number the topmost line in the group (e.g. D7) using the normal naming technique, described above, then hold Command and Option while clicking on successive lower-numbered lines, or
- number the bottom-most line in the group (e.g. D0) using the normal naming technique, described above, then hold Command, Option and Shift while clicking on successive higher-numbered lines.



Note that when you select each successive line, the new name appears, then it is necessary to click again or press Enter to make the name permanent.

Text and Picture Objects

Text and picture objects are used only to enhance the graphical appearance of a schematic diagram. They have no logical significance in the design.

Random text items are not associated with any particular device or signal on the screen and should not be used to set a name or attributes for devices or signals. The text in these boxes is not accessible in net or component lists. Use the naming and attribute features to attach text to devices and signals.

Creating a Text Block

If you click the pencil cursor on the diagram not near a device or signal line a blinking cursor will appear at that point and you will be able to type any desired text on the diagram. Text entry is terminated by the Enter key.

Editing random text

If you click the pencil cursor inside an existing text item the insertion point will be positioned at the click point. You can then use normal Macintosh text editing techniques to modify the text. Note that text on the clipboard can be pasted into an existing text box using the ⌘-V key equivalent for the Paste function. The Paste menu command will cause the current text entry to be terminated and a new text box to be created. Similarly, the ⌘ key equivalents for Cut (X) and Copy (C) can also be used while editing a text box.

Text boxes can be Zapped, Duplicated, Cut, Copied, Pasted and dragged just like any other item on the screen. See the descriptions of these commands for more information.

See Chapter V - Advanced Schematic Editing for more information on creating title blocks and sheet borders.

Sheet Sizes

DesignWorks 3.0 allows sheet size and border layouts to be set by one of two methods:

1) Importing an existing layout using the Import Sheet Info command. This is the normal method of setting the sheet size for a new design. DesignWorks is shipped with a number of templates for standard sheet sizes. These are actually just empty designs in which the sheet info has been preset. You can create your own custom sheet layouts which can then also be imported by this method.

2) Creating a custom layout using the Custom Sheet Info command. This command, and the associated text and picture capabilities, allows all aspects of the sheet size, drawing grid, border text style, title blocks and graphics to be set up to meet your drawing standards.

Custom Sheet Sizes

DesignWorks supports very flexible use of plotters and printers for output on a variety of media. For this reason, there can be complex interactions between the drawing area allocated for each page and the way the schematic is presented when it is output on various kinds of output devices.

The following information is considered to be part of the complete sheet definition:

- the overall height and width of the drawing area.
- the origin and spacing of the location grid.
- the format of text used in the border.
- whether or not the border size should be linked to the current printer page setup.
- whether or not the page should auto-expand to multiple sheets.
- any text and picture objects marked as "border" items.

When the Import Sheet Info command is used, all the above information is imported from page 1 of the selected design.

Default Sheet Layout

In this manual the term "page" is used to refer to a circuit page, which is the single, contiguous area that is viewed in one circuit window. The term "sheet" is used to refer to the physical sheet of paper that the diagram is printed or plotted on. Depending on the sheet size settings, a page may correspond to one or more sheets.

The default sheet size options "Use Page Setup" and "Single Sheet" give the simplest situation. With these settings, each circuit page will be printed on a single sheet of paper and any change in the printer Page Setup will be immediately reflected by a change in the drawing border.

When you first create an empty circuit using the New command, the drawing area defaults to a single sheet using the current printer Page Setup. Any area outside this sheet will appear in gray on the screen.

General Editing Concepts

The Clipboard

The standard Macintosh clipboard commands Cut, Copy and Paste can be used to move or copy circuit fragments, graphical and text information within a single circuit window, between windows, and between different programs (e.g. word processing or drafting).

Using Clipboard Data From Other Programs

When you start up DesignWorks, the clipboard may contain graphical or text information Cut or Copied from a document in another program. DesignWorks allows you to make use of this information in two ways:

- a Macintosh picture ("PICT" format) can be Pasted directly onto your diagram in any desired position. Once placed, it can be dragged, Zapped, Cut, Copied, Duplicated, etc. using the various editing commands available for other circuit elements. When a picture item is selected, the Get Info command can be used to select an optional border. See more information later in this chapter.
- text information from a word processor or text editor can be pasted into a text block. See more information under the Text command in the Edit menu.

Using Clipboard Data From DesignWorks

When a Cut or Copy is done, two types of data are placed on the clipboard:

- A Macintosh picture ("PICT" format) of the selected items, which can be pasted into a graphics document using most drawing programs.
- The DesignWorks circuit info for the selected items. This data is in a format that only DesignWorks can understand and is discarded when you Quit. This means that you cannot transfer circuit information between DesignWorks sessions.

Cut and Copy work on the currently selected group of objects and will be disabled if nothing is selected. See the earlier section in this chapter on "Selecting Circuit Objects". When items are copied onto the clipboard, their names are copied with them, which may result in duplicate names. If duplicate signal names are pasted back into the circuit page they were copied from, then logical connections will be made between the like-named segments.

See Chapter IX - Schematic Menu Reference for information on the specific clipboard commands.

Selection

Many DesignWorks commands, such as Get Info, Cut, Copy, etc., operate on the currently selected objects. For a detailed list of the techniques available for selecting objects, see the Editing Procedures section at the end of this chapter.

Multi-page Selection

The DesignWorks commands that operate on selected objects normally only affect objects on the current page. However, some external tools, such as the Browser, Find and ErrorFind, operate across pages and can select objects on non-current pages. This is done to allow global operations to be done without having to go to each page and perform separate editing operations. For example, Find can be used to locate all devices of a particular type and select them, then Browser can be used to display and edit their attributes.

Note the following rules for multi-page selection:

- Clicking in an empty part of a schematic window deselects all objects everywhere in the design.
- Switching from one schematic window to another does not change the status of any objects that were selected on either page.
- You can select objects on one page, switch to another page, and (using the normal SHIFT key technique) add items on the new page to the selected group.
- Editing commands like Cut, Copy and Clear will only affect selected items on the current page, even if objects are selected on other pages.

Schematic Editing Procedures

The purpose of this section is to give you a quick outline of commonly-used schematic editing techniques in DesignWorks. Background on each of these techniques is provided earlier in this chapter and in other reference sections of the manual.

Creating a New Design

A new design is created by selecting the New Design command in the File menu. The new design will consist of an empty master circuit with a single page. This page will appear in a circuit window entitled "Design1:1" (the

":1" portion indicates that this is page 1). The New Design command does not create a disk file. The design exists only in memory until you do a Save Design As command.

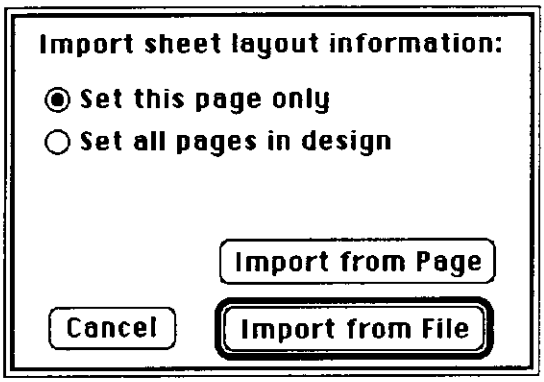
Your circuit diagram is created by first placing one or more devices in the circuit window (as described below), and then interconnecting the device pins with signal connections.

Disposing of a Design

DesignWorks has no built-in command to dispose of a design file. All information about a design is stored in a single file. Dragging the file to the Trash using the Finder has the effect of erasing all data concerning that design.

Selecting a Standard Sheet Size

Select the Import Sheet Info command in the Drawing menu. The following box will appear:



Click on the Import from File button. Locate the Sheet Templates folder inside the DesignWorks folder. Select the desired sheet template and click the Open button.

Navigating Around a Schematic Page

In addition to the standard scroll bars and Reduce/Enlarge menu items, DesignWorks has a number of convenient features for moving around a diagram.

Auto-Accelerating Scroll Bars

The scroll bars on a Schematic window operate in the same manner as all Macintosh applications with the following additional feature: holding the mouse button depressed in the scroll arrow at either end of a scroll bar will engage the "auto-accelerate" feature. Each scroll step will be larger than the previous one so that scrolling proceeds more and more rapidly in the direction of the arrow.

Auto-Scrolling

Whenever the mouse button is depressed and moves close to the edge of a schematic window, the window automatically scrolls to expose more area on that side.

Zoom (Magnifying Glass) Tool

The item in the tool palette is a powerful tool for moving around in a schematic diagram. It can be used to zoom both in and out and can control the exact area displayed on the screen.

1) Clicking and releasing the mouse button on a point on the diagram will zoom in to that point by one step.

2) Clicking and dragging the mouse down and to the right zooms in on the selected area. The point at which you press the mouse button will become the top left corner of the new viewing area. The point at which you release the button will become approximately the lower right corner of the displayed area. The circuit position and scaling will be adjusted to display the indicated area.

3) Clicking and dragging the mouse upward and to the left zooms out to view more of the schematic in the window. The degree of change in the scale factor is determined by how far the mouse is moved. Moving a small distance zooms out by one step (equivalent to using the Reduce command). Moving most of the way across the window is equivalent to doing a Reduce to Fit.

Find Tool

The find tool allows you to locate any device, signal or pin by name or any other attribute value. To use it:

- Select Find in the Tools menu.
- Select the type of object you wish to search for, device, signal or pin.

- Select the scope of the search, i.e. current page, current circuit level, entire design.
- Enter the name of the object to search for, or other search criteria.
- Click the Find button.

The Find tool automatically displays the first item found. The Next button can be used to cycle through all objects matching the given search criteria.

More detailed information on the Find tool is provided in Chapter XIII - Find and ErrorFind Tools.

Browser Tool

The Browser tool is a convenient way of viewing all objects in the design in tabular format. The Show button on the Browser panel can then be used to display any given object. See Chapter XII - Browser Tool for more information.

Placing and Editing Devices

Selecting a Device From a Library

To select a device from a library for placement in the schematic:

- Select the desired library using the pop-up menu at the top of the Parts palette.
- If necessary, use the scroll bar to scroll the list until the desired part name is in view.
- Double-click on the part name in the list.
- Move the cursor to the current schematic window.

The cursor will be replaced by a flickering image of the selected device. While moving this flickering image around, you can use the arrow keys on the keyboard or the orientation tool on the tool palette to rotate the symbol.

Clicking anywhere in the circuit diagram will make a permanent copy of the flickering device at that point.

NOTE: Holding the Option key while clicking will inhibit checking for pin connections. This allows you to select the device again and drag it to a new position without affecting any existing connections.

Duplicating an Existing Device

Either:

- select a similar device anywhere on the current circuit page and use the Duplicate command (either in the Edit menu or in the device pop-up menu).
- select a similar device in any other page of any open design and use the Copy command. Return to the destination circuit and select the Paste command.

After either of these operations, the cursor will be replaced by a flickering image of the selected device which can be placed by clicking in the schematic, as discussed above.

Deleting a Device

Devices can be removed by either of two methods:

- select the device by clicking on it (holding the SHIFT key if it is a switch or other input device) and then hit the "delete" or "delete" key on the keyboard, or select the Clear command from the Edit menu.
- enter Zap mode by selecting the Zap command or clicking on the Zap icon in the tool palette, then click on the device in question.

Moving a Device

Devices can be moved by clicking and dragging to the desired new position. If more than one device is selected, all the devices, and all signals connecting between them (whether or not selected) will be moved. Signal lines will be adjusted to maintain right angles at points where moving signal lines intersect with non-moving ones.

Entering Device Attributes

To enter device attribute, either:

- Hold the ⌘ key while clicking on the device, then select the Attributes command in the pop-up menu, OR,
- Select the device by clicking on it, then choose the Get Info command in the Options menu, then click the Attributes button.

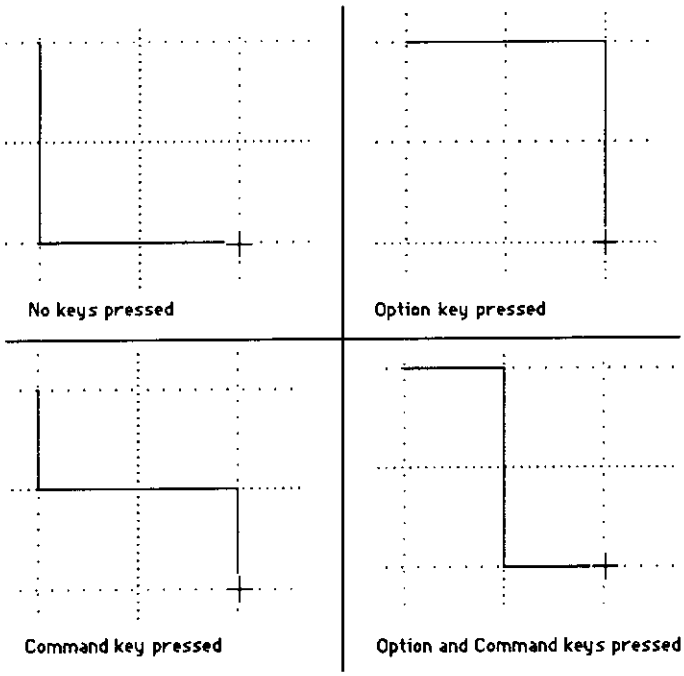
See more information on attributes in Chapter VI - Attributes.

Signal Line Editing Procedures

Drawing From an Existing Line or a Device Pin

A line can be extended from the end of an existing line or device pin using the normal ( ) cursor. Click and hold on the end of the pin and drag away from the pin. A pair of right-angle lines will follow the cursor away from the pin and long as the mouse button is pressed. Releasing the mouse button makes these lines permanent. If the end of the line (i.e. the point where the mouse button was released) touches another signal line, a connection will be made at that point.

Alternate line routing methods can be activated by pressing the Option and ⌘ keys, as follows:



The Shift key constrains the movement to a single vertical or horizontal line. The Option key inverts the order of line drawing, and the ⌘ key switches to three line segments with a center break.

NOTE: Holding the Option key while clicking will inhibit checking for pin connections. This allows you to select the signal again and drag it to a new position without affecting any existing connections.

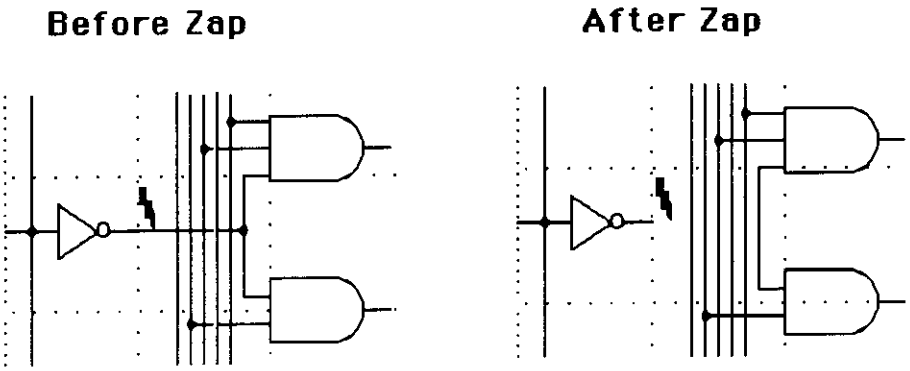
Creating an Unconnected Signal Line

The Draw Sig (+) cursor can be used to create an unattached signal line, or can be used to extend an existing signal. Simply click anywhere in the schematic and drag away in the desired direction. Unlike the Point mode drawing method, above, the mouse button does not have to be held while creating signals in this mode. Double-clicking terminates the signal line.

Editing a Signal Line

The following features are available to edit signal lines:

- Zap mode (entered by selecting the Zap command in the Edit menu or the Zap item in the tool palette) allows you to remove any single line segment from a signal connection. Zapping on a signal line removes only the line segment being pointed at, up to the nearest intersection, device pin or segment join point.



- selecting a signal line (by clicking anywhere along its length) and hitting delete or selecting the Clear command removes an entire signal trace.
- drawing backwards along the length of an existing line causes the line to be shortened to end at the point where you let the button go.
- clicking and dragging the middle of a signal line segment allows you to reposition the line. Vertical lines can be moved horizontally and vice versa.

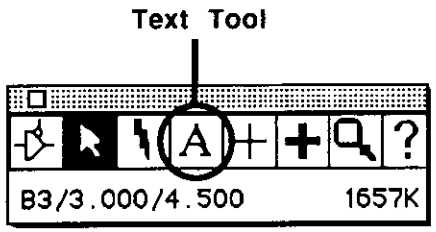
Checking Signal Interconnection

Double-clicking anywhere along a signal line will cause that signal segment and all logically connected segments on the current page to be selected.

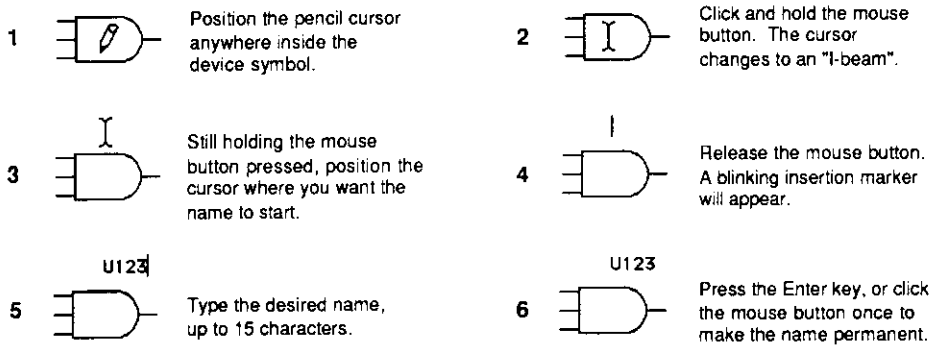
Name Operations

Adding a Device Name

Enter "Text" mode either by selecting the Text menu item in the Edit menu, or by clicking on the text icon in the tool palette:



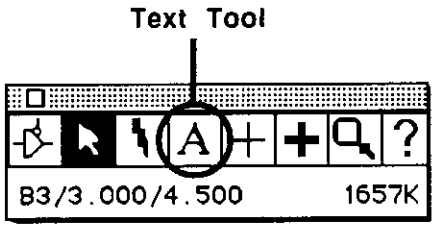
Note that once "Text" is selected, the cursor changes to a pencil icon. Press and hold the mouse button with the tip of the pencil positioned inside a device symbol. As long as you hold the mouse button down an I-beam cursor will track the mouse movements. The device name text will start at the position where you release the button. Type the desired name and press Enter, or click the mouse button anywhere. This sequence of steps is illustrated below.


1
Position the pencil cursor anywhere inside the device symbol.
2
Click and hold the mouse button. The cursor changes to an "I-beam".
3
Still holding the mouse button pressed, position the cursor where you want the name to start.
4
Release the mouse button. A blinking insertion marker will appear.
U123
5
Type the desired name, up to 15 characters.
U123
6
Press the Enter key, or click the mouse button once to make the name permanent.

Once a name is placed, it can be repositioned by dragging it using the Point cursor or removed using the Zap cursor. The device name will be removed automatically if the device is removed.

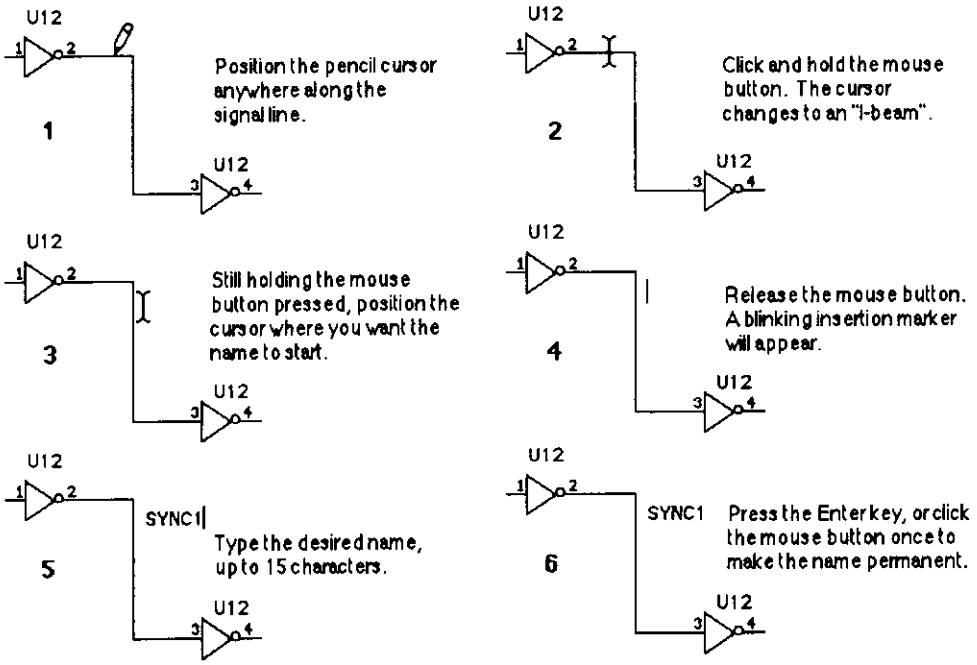
Adding a Signal Name

Enter "Text" mode either by selecting the Text command in the Edit menu, or by clicking on the text icon in the tool palette:



Note that once "Text" is selected, the cursor changes to a pencil icon.

Press and hold the mouse button with the tip of the pencil positioned anywhere along a signal line except within 5 screen dots of a device. As long as you hold the mouse button down an I-beam cursor will track the mouse movements. The signal name text will start at the position where you release the button. Type the desired name and press Enter or click the mouse button anywhere. This is illustrated below.


U12
Position the pencil cursor anywhere along the signal line.
1
U12
Click and hold the mouse button. The cursor changes to an "I-beam".
2
U12
Still holding the mouse button pressed, position the cursor where you want the name to start.
3
U12
Release the mouse button. A blinking insertion marker will appear.
4
U12
SYNC1
Type the desired name, up to 15 characters.
5
U12
SYNC1
Press the Enter key, or click the mouse button once to make the name permanent.
6
U12

Adding an Invisible Name

An invisible name for either a device or signal is created by either:

1) Holding the ⌘ key while clicking on the device or signal, then selecting the Name command from the pop-up menu.

OR

2) Selecting the desired device or signal, then selecting the Get Info command in the Options menu (⌘-I), then clicking on the Attributes button in this box, then selecting the Name field in the attributes box.

In either case, if the name is already visible on the diagram, changing it here will change all displayed occurrences of it.

Making an Invisible Name Visible

An invisible name can be made visible by simply clicking the Text cursor anywhere on the signal or device. When the mouse button is released, the name will be positioned at that point, as shown in the general naming instructions above.

Removing a name

A device or signal name can be removed by using the Zap cursor, described above. If the signal has been named in multiple locations then Zap removes the name only at the location zapped.

Editing a Name

The name can be changed by simply clicking the Name cursor on the signal name and editing using the keyboard as desired. Alternatively, a name can be edited by doing a Get Info on the signal. Changing the name in the Get Info box or at any one location on the diagram will change all visible occurrences of it.

Moving a name

A device or signal name can be moved by selecting the Point (normal) cursor, clicking and holding the mouse button on the name, and dragging it to the desired new position. Pin numbers cannot be repositioned.

Multiple Naming of Signals

A signal name can appear in up to 100 positions along the length of a signal line. To add a new position, simply use the normal naming procedure given in the section on signal naming i.e.:

- select Name mode
- click and drag anywhere along the signal line

- with the mouse button pressed, move to the desired position for the name
- release the mouse button

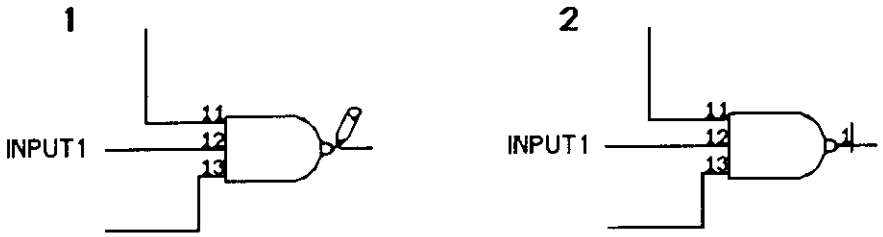
A new copy of the signal's name will appear at this point followed by a flashing cursor. To accept the name, simply click the mouse button once or press the Enter key. If you edit any occurrence of a name along a signal segment, all other occurrences will be updated to reflect the new name.

Any occurrence of a signal name can be removed using the Zap cursor. If you remove the last visible name from a signal segment then the logical connectivity to other like-named signals is removed.

Setting and Editing Pin Numbers

Editing Pin Numbers On the Schematic

If the mouse button is pressed with the tip of the pencil positioned on a signal line within 5 screen dots of a device, a blinking insertion bar will appear immediately where the signal joins the device. You cannot set the text position for pin numbers. Type the desired 1 to 4 character number and press Enter, or click the mouse button anywhere, to make the number permanent. See the sequential numbering option discussed below under "Auto-naming Features".



Editing Pin Numbers Using Get Info

To edit pin numbers using the Get Info dialog box:

1) Click on the device in question to select it.

2) Click on the Pin Info box. This will display the pin information for the first pin.

3) Edit the pin number as desired.

4) Click the Next Pin button to see the next pin in the list.

Editing Pin Numbers Using the Browser

To edit pin numbers using the Browser tool:

1) Select the Browser command in the Tools menu.
2) Select the Pins object type.
3) Select the Number item in the Pseudo pop-up menu.
4) Select the Edit item in the Mode pop-up menu.

Pin numbers can now be edited in the "Number" column of the spreadsheet.

Selecting Circuit Objects

To select circuit objects, the cursor must be in the normal pointer ( ) mode.

Selecting a Device

A single device is selected by clicking the mouse button with the pointer positioned anywhere inside the device symbol or in any displayed attribute value associated with the symbol.

Simulated input devices such as switches and keyboards can only be selected by holding the SHIFT key while clicking. This is necessary because a normal click is used to change the state of these devices when a simulator tool is active.

Selecting a Picture or Text Object

A single, non-background picture or text item is selected by clicking the mouse button with the pointer positioned anywhere inside the item.

Selecting a Background Object

To select a picture or text item marked as "background" you must hold the Command and Option keys pressed while clicking on it.

Selecting a Signal

A single signal is selected by clicking anywhere along the signal line. This selects only the part of the signal directly attached to the clicked line.

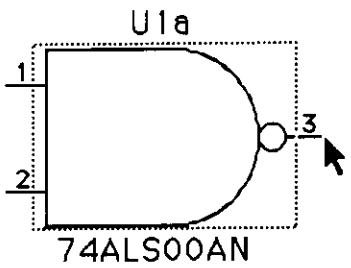
Double-clicking the signal selects all parts of the signal on the given page including logical connections by name or bus.

Selecting a Pin

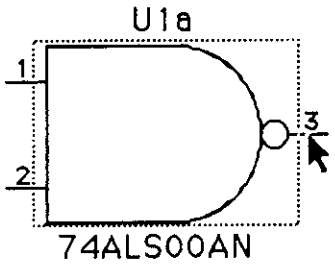
A pin is selected by clicking on the pin line close to the device. NOTE: Since an unconnected device pin is both a pin and a signal, you determine whether you get the pin or signal pop-up menu as follows:

- clicking on the pin in the last 1/4 of the pin length away from the device will display the signal menu.
- clicking on the pin close to the device symbol will display the pin menu.

Selecting the Signal



Selecting the Pin



Selecting Groups of Objects

Several methods are available for selecting multiple objects:

1) Any group of adjacent items can be selected by clicking and dragging across the group. A flickering rectangle will follow the mouse movement. Any object that intersects this rectangle when the button is released will be selected (except background objects).

2) A group of interconnected devices and signals is selected by double-clicking on any device in the group while holding the Option key pressed. If a circuit is completely interconnected, this will select the entire circuit.

3) The Select All command in the Edit menu selects all items on the current page except background items.

4) The SHIFT key can be used in combination with any of the above methods to select multiple items. When the SHIFT key is held, the previously selected items are not deselected when a new item is clicked on.

Thus you can add to the selected group until the desired collection of items is selected.

Changing Search Order

Holding the Option key pressed while clicking the pointer causes object types to be searched in the opposite order from normal. This can be used, for example to select a signal name that has accidentally moved under a device, etc.

Deselecting a Selected Object

All currently selected objects are deselected by clicking in an empty area of the schematic window. A single item can be deselected by holding the SHIFT key while clicking on it.

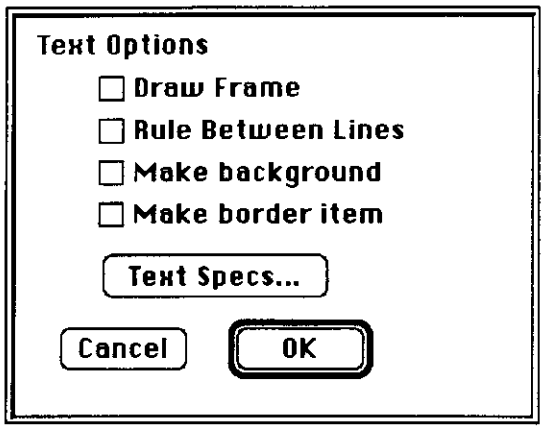
Creating a Text Notation

If you click the text cursor on the diagram not near a device or signal line, a blinking cursor will appear at that point and you will be able to type any desired text on the diagram. The Return key can be used to enter multiple lines in a single text block. Text entry is terminated with the Enter key.

NOTE: Random text items are not associated with any particular device or signal on the screen and should not be used to name or set attributes for devices or signals. The text in these boxes is not accessible in net or component lists. Use the naming and attribute features to attach text to devices and signals.

Text Style and Display Options

To set text display options and text style, select the text block by clicking on it with the Point cursor, then select the Get Info command in the Options menu. This will display the following box:



The following table summarizes the options available in this box.

Draw Frame	Turning this switch on causes a frame to be drawn around the text item on the schematic.
Rule Between Lines	Turning this switch on causes a line to be drawn after each row of characters.
Make Background	Turning this switch on makes the selected text block a background item, i.e. it cannot be affected by normal editing operations. In order to select it for editing, hold the Command and Option keys pressed while clicking on it.
Make Border	Turning this switch on marks this item as a . This means that it will be treated as part of the sheet border. I.e. it will be updated by the Import Sheet Info command, copied to new pages as they are created, and unaffected by the Center in Page command.
Text Specs	Clicking this button displays the standard text style dialog. Any changes made in text style affect only the selected item, but they also become the default for future text blocks.

See more information on using text notations in title blocks and sheet borders in Chapter V - Advanced Schematic Editing.

Creating a Picture

Pictures can not be created from scratch right on the schematic diagram. They can be created either in another application or using the DevEditor tool. To do this:

- enter the DevEditor tool, or other paint, drawing or drafting program to create the graphics object.
- select the desired object.
- select the Copy command in the Edit menu. This places a copy of the graphics item on the clipboard.
- select your DesignWorks circuit.
- select the Paste command from the Edit menu. A flickering image of the graphics object will follow the mouse movements until you click once in the desired position.

This page is intentionally left blank.

Chapter V - Advanced Schematic Editing

This chapter provides information on the more advanced schematic editing features of DesignWorks.

Device Packaging

DesignWorks 3.0 provides features for automatic or manual assignment of devices to physical packages. This mechanism is intended for use when creating designs that are destined for production in printed circuit board (PCB) or other discrete package format.

The Packager tool has the following features:

- Automatically generates a unique package assignment for each device placed on the schematic.
- Uses a name prefix stored with the device symbol, or a default value stored with the design. This allows appropriate names to be generated for discrete parts, e.g. "D1", "D2", "D3", etc. for diodes, and "R1", "R2", "R3", etc. for resistors.
- Automatically handles device types having multiple units per package, e.g. gates. Special attributes are stored with the symbol to tell the packager how many units are available and the pin assignments for each.

Packaging vs. Auto-Name Assignment

DesignWorks has two related features which can both assign names to devices: Packaging and Auto-Name Assignment.

- Packaging assigns a name, unit and pin numbers to each device based on physical packaging considerations. Packaging is controlled by the commands in the Packaging sub-menu in the Options menu.
- Auto-naming is more restricted and simply ensures that each device symbol in a circuit has a unique name. It is controlled by buttons in the Design Preferences box.

In flat hierarchy mode (the default), packaging takes precedence, i.e. if auto-packaging is enabled, auto-name assignment will have no effect. In physical hierarchy mode, both features can be used together, since the package assignment is not normally used as a device name in hierarchical designs. Packaging is not available in pure hierarchy mode.

Auto Packaging

When Auto Packaging is enabled (i.e. the Auto Packaging Enabled menu item is checked), the Packager assigns each device to a package as soon as it is placed on the diagram. When a device is deleted, its package assignment becomes available for the next device of the same type that is created. In this mode, package assignments will be in chronological order of placement on the diagram.

It is normally convenient to leave Auto Packaging turned on while initially creating a design. This way, all devices get reasonable initial name assignments. The design can be repackaged later (using the Repackage Design command) once the physical layout of the design becomes more clear.

Re-enabling Auto-Packaging After Manual Edits

If Auto-Packaging has been disabled while manual editing was done, then the design's package table will be out of date. It may be desirable to select the Rescan Design command from time to time to recheck the validity of the current package assignments. This command brings the package table up to date based on name and unit assignments currently in the design, without changing any values. In any case, Auto-Packaging cannot be re-enabled until a successful Rescan is completed.

Auto-Packaging Limitations

The Packager can be confused by some conditions when package assignments are changed manually. Such conditions include:

- two devices given the same packaging assignment.
- a name or unit attribute field set to an invalid value.
- Packager attributes changed.

In these cases, the Packager will display a warning box and disable auto-packaging. The error must be corrected or the offending device marked with the "Ignore" option (see below) before auto-packaging can be re-enabled.

NOTE: The Packager looks only at the name and unit attribute fields, not at the pin numbers themselves. If you manually change any pin numbers on devices without making corresponding changes to the Unit field, invalid package assignments could result.

Batch Packaging

The Repackage Design command in the Options menu allows you to request a complete reassignment of device packages. This command performs the following steps:

- the design package table is cleared.
- all devices marked as Locked (see below) are noted in the package table and checked for errors.
- all unlocked devices are sorted and assigned to packages.

For packaging, devices are sorted by page number, then by position on the page. Names are assigned in vertical strips corresponding to the lettered grid references on the page. I.e. all devices in column A are assigned, starting at the top of the page and working down. Next, all devices in column B are assigned, etc.

Manual Packaging

All aspects of device packaging can be controlled manually, if desired. This can be cone on two levels.

Manual/Automatic Packaging

The higher-level approach takes advantage of the packaging information in the device libraries and the error checking capabilities of the Packager. The following packaging features can be used:

- Each library entry contains information about available gate package assignments. The list of available units appears in the Unit sub-menu of the device pop-up menu. Selecting one of these items automatically updates the Unit field and the pin number assignments on the device.
- The Browser tool can be used to view and change all name and unit assignments for the design. Changing the Unit value from the Browser will automatically update the pin numbers.
- The Rescan Design command can be used to check for packaging errors without affecting any existing settings. Any duplicate assignments or invalid unit settings will be announced.

The following cautions should also be noted:

- The Packager does not check pin numbers when checking package assignments. It uses only the contents of the Unit attribute field. If you manually change any pin numbers on devices without making corresponding changes to the Unit field, invalid package assignments could result.

• It is best to set the Packaging Level to Locked (using the Get Info command) for devices that you wish to have a fixed package assignment, even if not using auto-packaging. This will avoid loss of data if auto-packaging is inadvertently enabled or a Repackage Design is done.

Fully Manual Packaging

The lower-level approach completely ignores all packaging features. This can be used if unusual pin assignments are needed, or libraries without packaging information are being used. Note the following points when doing manual packaging:

• Pin numbers can be edited directly on the diagram or assigned using the Pin Info box or the Browser tool.

• If you do not wish to display a unit assignment (e.g. "a", "b", "c", etc.), you can use the Define Attribute Fields command to turn off the Visible by Default option for the Unit field.

• If you wish to manually assign a unit value, it is best to create a user-defined attribute field (e.g. "GateUnit") and make all entries in that field. This is necessary to circumvent the automatic updating of pin numbers when the Unit field is changed.

• When a netlist or Bill of Materials is created using the Report tool, all devices with the same name are normally treated as a single device. This means that they are, in effect, assigned to the same package. The Report tool does not check for duplicate pin numbers or name assignments, although it is possible to customize a report format that will assist in locating these problems.

Auto-Generated Name Format

Device names are generated by appending a numeric suffix to a prefix provided. If the prefix field (i.e. the field whose name is in the design's PrefixField field) is non-empty in the device, then it is taken as the prefix. This is normally set using the DevEditor tool in the device library entry, although it can be set individually for any device on the schematic. If the prefix field is empty, then the contents of the design's "PkgPrefix" field are used.

NOTES:

1) The default prefix field is Name.Prefix. This can be changed by entering the name of any other field in the design's PrefixField field.

2) Most library parts provided with DesignWorks that represent logic devices or integrated circuits have no Name.Prefix value. They will therefore use the value of the design's PkgPrefix field, "U" by default. Discrete parts or others that normally have specific standard prefixes are set to the common values.

The numeric suffix is assigned sequentially for each different prefix found in the design. E.g. the packager will assign names "U1", "U2", "U3", etc. to devices with the value "U" in their prefix field, and names "R1", "R2", "R3", etc. for devices having prefix "R".

Device Packaging Level

DesignWorks allows you to mark individual devices for special handling during packaging operations. Using the Get Info command, you can select one of the following packaging levels for each device in the design:

Normal (the default value)	This device will get a package name and unit number assigned automatically by the Packager. The assignment can be changed by future repackaging operations if necessary.
Lock and Check	This device's package assignment will not by changed, but it is recorded in the package table so no other device will be assigned the same name. The device will be checked for packaging errors.
Lock and Don't Check	The same as Lock and Check except that the package assignment is not checked for errors. This could be used in special cases where the same name is purposely assigned to two different symbols.
Ignore	This device is completely ignored by packaging operations. I.e. its name and unit will not be changed, no error checking is done, and some other device might be given the same name.

Back Annotation and Packaging

Back annotation refers to any process that reads a list of name, unit and pin numbering changes from an external system and automatically updates the design file. Obviously, any such process can potentially bypass the Packager and create invalid package assignments. It is recommended that the

Rescan Design command be used after any such process to ensure that the name and unit assignments are still valid.

> WARNING: The Rescan Design command does not guarantee the correctness of your design. Pin numbers are not checked. Check your design carefully after any Back Annotation process.

Packaging in Hierarchical Designs

Hierarchy Mode

The usage of attribute fields by the Packager is different between flat and physical hierarchy modes. See the discussion below under "Packaging Attribute Fields". Packaging is not available in pure hierarchy mode.

Name vs. InstName

In hierarchical designs, a device symbol in a sub-circuit can represent multiple physical devices. This happens if the parent device has been instantiated multiple times in the design. For this reason, the device "name" is split into two different functions in a hierarchical design.

The "Name" attribute field is associated with the definition of the sub-circuit and will be the same for all instances of the sub-circuit. For this reason, it can be thought of as the "logical name" or "definition name" of the device and cannot be used to hold the package assignment.

The "InstName" attribute field is associated with each instance of the sub-circuit. It can take on a different value for each physical device it represents and is used to store the package assignment in hierarchical designs. The user can choose to display either or both of these names on the schematic.

See Chapter VII - Hierarchical Design for more information.

Restricting Packaging Depth

In a hierarchical design, a device symbol that represents a physical device may have a sub-circuit for simulation or analysis purposes. Normally, the parent device rather than the contents of this sub-circuit should be packaged. The packaging options in the device Get Info command can be used to control packaging of internal circuits. Standard library devices provided with DesignWorks will default to packaging the parent device even if an internal circuit is added.

Name Assignment Order

In hierarchical designs, all packagable devices in a given circuit level are assigned before the internals of any sub-circuit devices in that circuit. This ensures that a given circuit level contains sequential device names and units.

Using Device Libraries Without Packaging Information

If the Packager encounters a device without any packaging attribute fields define, it takes the following minimum action:

- the device is assumed to have only one unit per package, i.e. each device will be given a unique name.
- the default package name prefix for the design will be used to assign a name.
- no unit assignment will be displayed on the schematic.
- no Unit selection menu will appear.
- default pin numbering defined in the library will be used.

Packaging Attribute Fields

Fields Set By the Packager

The Packager stores the package name and unit assignments in attribute fields associated with each device symbol. The fields that are used depend upon hierarchy mode:

Flat Mode	The package name is stored in the "Name" field and the gate unit (e.g. "a", "b", "c", etc.) is stored in the "Unit" field. This reflects the common usage that the device name used on the schematic is the name of the package on the final PCB.
Physical Mode	The package name is stored in the InstName field and the gate unit is stored in the Unit field. The Name field cannot be used because a single device in a sub-circuit may actually represent multiple devices on the physical PCB. Each one of these devices requires a separate package assignment. The "InstName" field can take on a different value for each instance.

Pure Mode	Packaging is not supported in Pure mode. This is because there is not instance data and therefore no place to store different package assignments for multiple instances of the same device.

Fields Read By the Packager

The following information is used by the Packager in determining what units are available in each package and pin assignments belong with each unit. Devices with only one unit per package do not need any special attribute fields.

Part	The Part field is used to determine which devices can share a package. In some devices, several different types of elements exist in one package. In this case, a different symbol is used for each different logic type, but each must have the same Part name.
Name.Prefix	This device attribute field contains the prefix used to generate the device name. If it is empty, then the design's PkgPrefix field is used. This field is normally defined in the device library entry, but can be overridden in each device for special requirements. NOTE: Name.Prefix is the default field used for this purpose, but this can be overridden. See the PrefixField description below.
PkgLevel	This field contains the package level setting for this device instance. If the field is empty, or contains the character "0", Normal mode is assumed. "1" means Lock and Check, "2" means Lock and Don't Check, "3" means ignore. Any other value will be interpreted as Lock and Check. This value is not normally defined in the library entry for the device. The package level is usually set using the buttons in the Device Get Info box, but the attribute field can be set directly if desired.

Unit.List	This device attribute field contains a list of the units available for this device symbol. The unit names are separated by commas, e.g. "a,b,c,d". In device types having different types of logical units in one package, this is the list of only those units which can be used with this symbol. This field is normally defined in the device library entry, but can be overridden in each device for special requirements.
Unit.All	This device attribute field contains a list of all units available in this package, whether or not they apply to this symbol. Normally this is the same as Unit.List, but will be different for devices having more than one type of logical unit per package. Format is the same as for Unit.List. This field is normally defined in the device library entry, but can be overridden in each device for special requirements.
PkgPrefix	This design attribute field contains the default package name prefix used if no prefix field appears in the device.
VisPin.List	This pin attribute field is used to set the pin number corresponding to each gate unit. It consists of a list of pin numbers separated by commas, e.g. "1,4,10,13". The order of the pin numbers in this list must match the order of the units in Unit.List. This value is normally defined for each pin in the device library entry, but can be overridden on the schematic if desired.
PrefixField	This design attribute field contains the name of the field to use for a device name prefix. The default is Name.Prefix, but this can be overridden to use specialized prefix values.

Device and Signal Auto-Naming

When Auto-Naming is enabled, all devices (except pseudo-devices) and signals will be given a default name when they are created. This ensures that every object in a design has a unique name and can be identified in a netlist.

Device Auto-Naming vs. Packaging

If the design is in flat hierarchy mode (the default) and auto-packaging is enabled (the default), then Device Auto-Naming will have no effect. In "flat" mode, both the Packager and the Device Auto-Naming feature set the Name attribute field. In this case, the Packager takes precedence.

If the design is in physical hierarchy mode, then the Packager sets the InstName field and the Device Auto-Naming sets the Name field.

> *See the comments on Name vs. InstName in hierarchical designs earlier in this chapter and in Chapter VI - Attributes.*

Enabling Auto-Naming

Auto-Naming is enabled by selecting the Design Preferences command in the Drawing menu. Checking the corresponding box causes all devices placed in the design subsequently to be given a default name if they are not already named.

If the "Visible" option is selected, the default name will be displayed on the diagram adjacent to the device or signal when it is created.

How Names are Generated

The auto-generated name consists of two parts, the fixed prefix and the numeric suffix. The prefix portion is derived from the device's prefix attribute field, if any, or from the design's DevPrefix field. The latter is normally set using the Design Preferences command. The standard prefix field, "Name.Prefix", is empty for most digital parts in the DesignWorks libraries. For discrete components, a commonly-used prefix letter is specified. For signals, the design's SigPrefix attribute field value is used to generate the name.

The numeric portion of the name is generated from the device or signal's "token" value. This value is guaranteed to be unique within a circuit level,

but no attempt is made to fill in unused values. Thus, in a design that has been edited, sequential numbering is not guaranteed.

Selecting an Alternate Prefix Field

The prefix used in generating device names is derived another device field known as the "prefix field". The default prefix field is "Name.Prefix". DesignWorks allows the prefix field to be selected for a given design so that multiple prefixes for different naming conventions can be stored with the same part. E.g. the standard Discretes library included with the package has two prefix fields:

- "Name.Prefix" is the normal one used by default for generating names.
- "Name.Spice" contains prefix values specific to SPICE-based analog simulators.

In addition, the standard field "Function" contains a short function code for the part that can also be used as a descriptive prefix.

The selected prefix field is changed by using the Set Design Attributes command to place the name of the desired prefix field in the PrefixField attribute field. See the procedures section later in this chapter for specific steps.

Device and Signal Token Values

Every time a device or signal is created in a DesignWorks circuit, it is assigned an integer value known as its "token". The token number stays with the device or signal for its lifetime and numbers are not re-used. This ensures that a given device or signal can always be recognized despite duplicate names or name changes. The token is used for a number of internal operations in DesignWorks, but can also be seen by the user in the following circumstances:

- The token number is used to generate default names for devices and signals, as described elsewhere is this chapter.
- The token number can be written out in netlists or bills of materials whenever a guaranteed-unique identifier is needed.

Note the following characteristics of tokens:

- Tokens are assigned independently for each circuit in a hierarchical design and are thus only unique within a circuit, not across the entire design.

• When two signals are joined, one of the two tokens is discarded at random.

• Each logical symbol on the diagram (including pseudo-devices) has its own token. In a netlist, several symbols may be combined into a single package, so there is not necessarily a one-to-one correspondence between tokens and physical packages.

Multipage Schematics

DesignWorks allows a circuit to be represented on multiple pages with logical connections between pages. When a new circuit is created, it is assumed to have a single page. A new page is added to the circuit by selecting the Page command in the Drawing menu and clicking on the New Page button.

Sheet Border Setup for Multi-Page Designs

When a new page is added to a circuit, all border information and text and picture objects marked as "border" items are copied automatically from the preceding page (i.e. what was formerly the last page). For this reason, it is most convenient to set up the first page of the circuit with the desired border arrangement before adding other pages.

Borders on all pages in a design can be updated later if needed using the Import Sheet Info command.

Inter-page Connections

Using the Page Connector Device

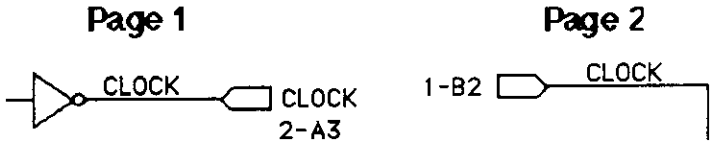
Signals drawn on different pages will initially have no logical connection, even if the names are the same. Inter-page connections are made using the Page Connector pseudo-device which is in the Primitives and Connectors libraries. When attached to a signal line, the Page Connector makes the name of that signal known across all pages, i.e. any like-named signal on another page which also has a page connector will be logically connected to this one. The Page Connector can be placed anywhere along the signal line although it is normally convenient to place it at the edge of the page. The Page Connector itself can be named (in order to take advantage of the automatic page references mechanism described below), but it must have the same name as the attached signal. Changing the name of the Page Connector (using the naming procedure for devices) will also change the

name of the attached signal. Likewise, changing the name of the signal will rename any attached Page Connectors.

Automatic Display of Page References

DesignWorks has an Automatic Page Reference facility which displays the page number and position of any Page Connectors associated with a given signal. These page references are automatically displayed immediately adjacent to any Page Connector as soon as multiple Page Connectors are attached to the same signal.

For example, in the following case both pages have signals named CLOCK with Page Connectors attached, so the two CLOCK signals are logically connected. Note that both Page Connectors are notated with the page number and grid reference:



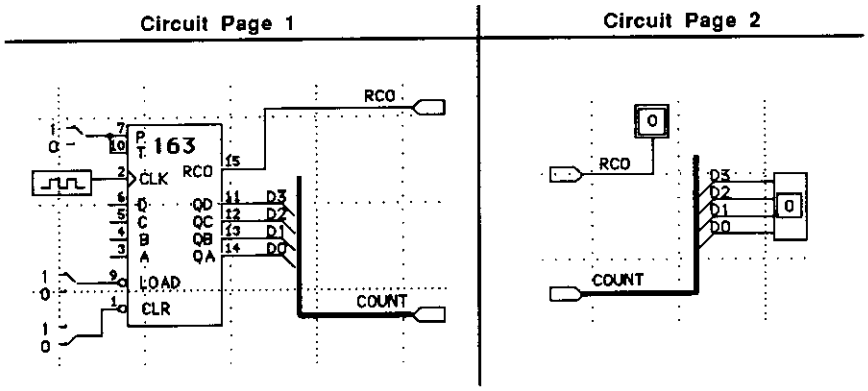
When a name is applied to the Page Connector itself, the page reference will appear under the name.

As long as automatic updating is enabled, these references will be updated automatically when any Page Connectors or attached signals are added, deleted, moved or renamed.

See the Design Preferences command in Chapter IX - Menu Reference for page reference format options and update control.

Connecting Busses Across Pages

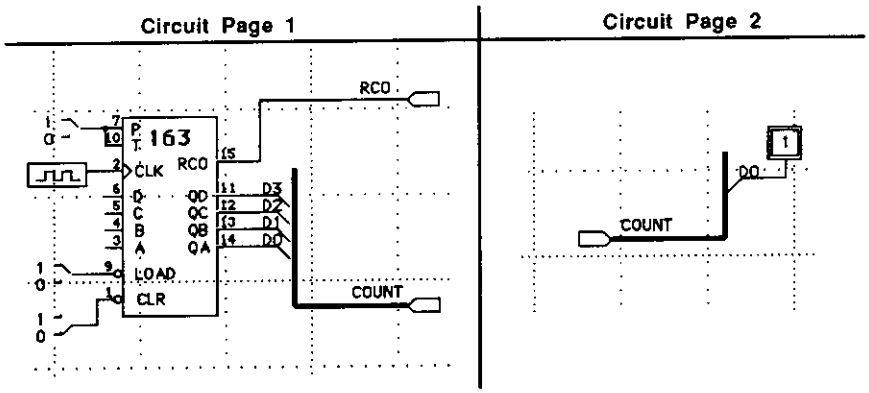
Busses can also be connected between pages by the same method, except that the Bus Page Connector must be used in place of the Page Connector. This is illustrated in the following simple circuit:



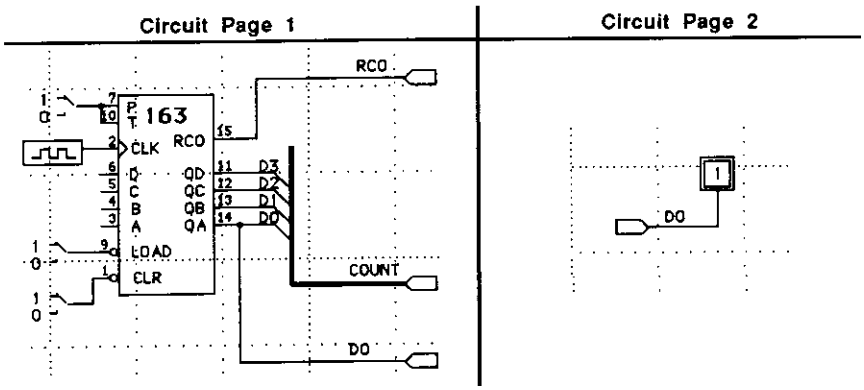
Page Connectors on Internal Bus Signals

Note that a Bus Page Connector does not make the names of the internal signals known globally, only the bus itself. To bring an individual internal bus signal across to another page, one of the following methods can be used:

1) Make the entire bus global using a bus page connector, then use a breakout to access the desired signal, as in the following:



2) Place signal page connectors on the individual signal on both pages, as in the following:



Changing the Page Connector Symbol

Note that the symbol for the Page Connector device can be changed using the DevEditor tool just as with any other device.

> *See Chapter X - Device Symbol Editing for specific information on creating Page Connector symbols.*

Bussing

The bussing facility allows any combination of named signals to be represented by a single line and any subset of these to be brought out through a "breakout" at any point along the bus line.

Properties of Busses

A bus is treated by DesignWorks as a signal with special properties. Thus, bus lines can be drawn and modified on the screen using all the same editing features available for signals. Note the following properties of busses:

- Only bus pins on devices can be connected directly to a bus. All other connections must be made by using a breakout to access the desired internal signals. A breakout is created using the New Breakout command in the Options menu.
- You do not need to specify in advance what signals will be contained in a given bus. Any signals that are present in a breakout or bus pin

attached to a bus will become part of that bus and can be brought out through another breakout anywhere along the bus.

• Any two busses can be joined together, regardless of their internal signals. When two different busses are merged, any signal in either bus becomes available anywhere along the combined bus.

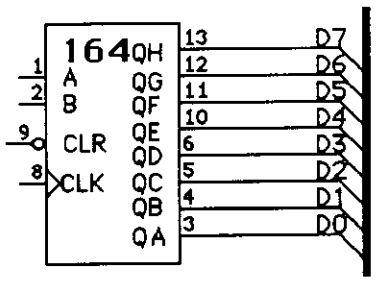
• If you select a bus line, then do a Get Info command from the Options menu the displayed info box will show a list of the signals currently contained in the bus.

• A given signal can be present only in one bus. If you attempt to connect together two signals in different busses, a warning box will be displayed and the connection will be canceled.

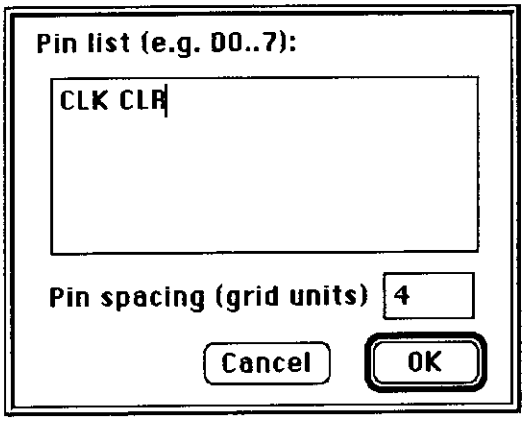
• A bus can be created by drawing the bus lines first, then creating the breakouts to attach, or by creating a breakout and extending the bus line starting at the bus pin. Bus lines are drawn or extended using exactly the same techniques as for signals, except that the Draw Bus command or cursor is used instead of Draw Signal.

Properties of Breakouts

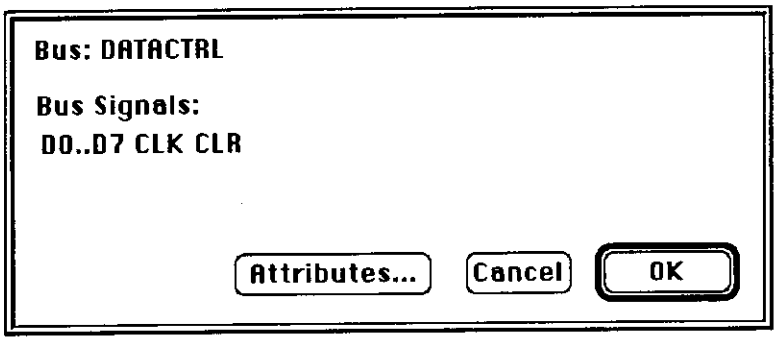
Signals are attached to a bus via a special type of device symbol called a "breakout". It is not legal to attach a signal line directly to a bus line and any attempt to do so will elicit a warning box. In DesignWorks, a breakout is treated as a device with certain special properties. This means that it can be placed in any desired orientation, moved, duplicated, etc. using any of the device editing features available. A typical breakout appears as follows:



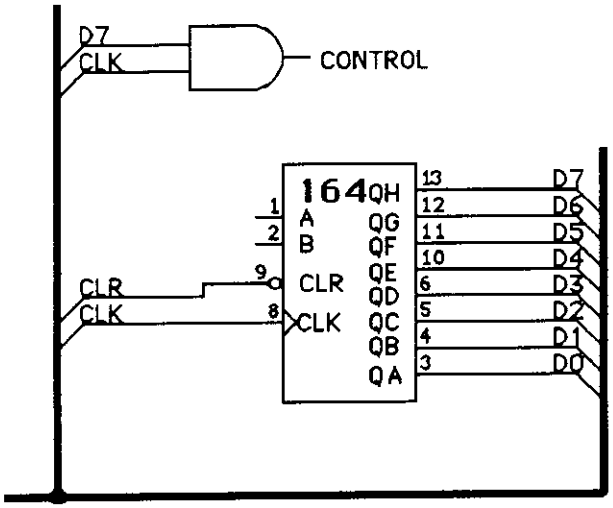
Any breakout can always be attached to any bus. When a breakout is attached that contains signals unknown in that bus, the signals are implicitly added to the bus. For example, suppose we want to add control signals to the above circuit, we could create a breakout containing only the new signals, as follows:



Once such a breakout has been added to the bus, all signals in all attached breakouts are considered part of that bus. A list of internal signals can be seen by selecting the bus and using the Get Info command:



Any combination of the internal signals can now be brought out of the bus at any point, as in the following addition to the above circuit:



Bus Pins

DesignWorks supports user-created bus pins on devices. A bus pin can be defined to have any collection of named internal pins. Note the following properties of bus pins:

- The bus pin itself does not represent a physical device pin. It is only a graphical place-holder on the schematic representing a group of internal pins. The bus pin itself never appears in a netlist.
- The internal pins represent physical device pins. Even though they do not appear on the schematic, they can have all the same parameters as normal devices pins, including pin numbers and attributes. These parameters can be accessed using the Bus Pin Info command in the pin pop-up menu.
- When a device with a bus pin is placed, it has a pre-created bus attached to it by default. This bus will contain one signal for each internal pin, with the initial name of the signal being the same as the name as the pin.
- A "splicing" box can be displayed using the Bus Pin Info command in the pin pop-up menu. This box allows any internal pin to be connected to any signal in the attached bus.

For more information on creating device symbols with bus pins, see Chapter X - Device Symbol Editing.

More on Text and Picture Objects

Background and Border Objects

Either text or picture blocks can be set to be "border" or "background" objects. These two options will normally be used together, but can be enabled separately if desired. These two options are enabled by selecting the object in question and using the Get Info command.

Border Objects

If a text or picture is marked as a "border" item, then it is considered to be part of the sheet border information for the page it is on. This has the following effects:

- if a new page is added to the circuit using the Page command, this item will automatically be copied to the same position on the new page.
- if the page containing this item is used as the source for an Import Sheet Info operation, this item will be copied to the destination.
- if a Center in Page command is used, this object will not be repositioned since it is assumed to be part of the border.

Background Objects

Text or picture items marked as "background" objects cannot be selected, modified or deleted using normal circuit editing techniques. This is used to prevent custom sheet backgrounds or border graphics from interfering with circuit editing operations. This is normally used in combination with the "border" status discussed above.

To select a background object, hold the Command and Option keys pressed while clicking on it. The Get Info command can then be used to disable the background status.

Using Text Variables

The text facility allows special entries called variables to be placed in text boxes. When the text is displayed or printed, the appropriate system quantity or attribute field value is substituted for the variable name. Any variable name that is not recognized will be displayed verbatim. This facility is intended to assist in creating title blocks and similar additions to the schematic with information such as the date, page number and title, revision level, engineer's name, etc. Using variables will allow this

information to be entered in one place so that changes are reflected automatically on all schematic pages.

System Variables

System variables start with a "$" and must be one of the following:

$DATENOW	This will be replaced by the system date written mm/dd/yy. This is the current date as maintained by the Macintosh clock.
$TIMENOW	This will be replaced by the system time written hh:mm. This is the current time as maintained by the Macintosh clock.
$DATECREATED	This will be replaced by the date the file was created, as maintained by the Macintosh file system.
$TIMECREATED	This will be replaced by the time the file was created, as maintained by the Macintosh file system.
$DATEMODIFIED	This will be replaced by the date the file was last modified, as maintained by the Macintosh file system.
$TIMEMODIFIED	This will be replaced by the time the file was last modified, as maintained by the Macintosh file system.
$PAGENUM	This will be replaced by the number of the page (within the current circuit level) the text is on.
$NUMPAGES	This will be replaced by the total number of pages in the circuit level.
$PAGETITLE	This will be replaced by the title of the page the text is on, as entered using the Page command.
$PRINTPAGENUMBER	This will be replaced by the page number in the printed sequence within the current Print Design command. This is used to number pages within a hierarchical design. NOTE: This is only valid for printed output. When it is drawn on the screen, it will show the same value as $PAGENUM.

$PRINTNUMPAGES	This will be replaced by the number of pages that will be printed by the current Print command. This is used to number pages within a hierarchical design. NOTE: This is only valid for printed output. When it is drawn on the screen, it will show the same value as $NUMPAGES.
$FILENAME	This will be replaced by the name of the design file.

Attribute Variables

Attribute variables start with a "&" mark and are used to refer to fields stored in the attributes for the design. These can be used to place information at multiple points on a diagram which will be updated automatically when the design attributes are changed. For example, if the field REV was defined for the design with the following contents:

```
2.1A Mar 18, 1993
```

then the variable &REV would appear as "2.1A Mar 18, 1993" on the diagram.

See the Set Design Attributes command in Chapter IX - Menu Reference.

Editing Text with Variables

Text items on the schematic will normally be displayed with variables replaced by their values. When a text item is selected or clicked on with the Name cursor it is redisplayed in its raw format with the variable names shown without interpretation. This allows the items to be edited with the text in its actual stored position.

Text Frame Size with Variables

The framing rectangle for a text item is calculated after the variable substitution is done. This may cause the item to be highlighted or deleted incorrectly if the variable values are shorter than the names. These errors are not serious or permanent and will disappear when the screen is updated. The text box can be expanded as necessary by adding blanks at the end of any one line in the item.

More on Sheet Sizes

Custom Sheet Sizes

DesignWorks supports very flexible use of plotters and printers for output on a variety of media. For this reason, there can be complex interactions between the drawing area allocated for each page and the way the schematic is presented when it is output on various kinds of output devices.

The following information is considered to be part of the complete sheet definition:

- the overall height and width of the drawing area.
- the origin and spacing of the location grid.
- the format of text used in the border.
- whether or not the border size should be linked to the current printer page setup.
- whether or not the page should auto-expand to multiple sheets.
- any text and picture objects marked as "border" items.

When the Import Sheet Info command is used, all the above information is imported from page 1 of the selected design.

Power and Ground Connections

Power and Ground Connections in Attributes

When a schematic diagram is created, the power and ground connections for devices are normally not drawn as signal connections on the diagram. These connections would clutter the diagram and are not necessary for an understanding of the logical function of the circuit. Obviously, though, these connections must be included at some level in order to form a complete netlist. Three methods are available to do this:

- The power and ground pin connections can be specified as attributes for each device. The Schematic tool itself attaches no significance to the attribute fields, but the report generator reads them and adds the specified pins to the netlist. The libraries of standard types supplied with DesignWorks include this information for each type. It is only necessary for you to enter this data if you are making non-standard connections or using a device that has no default power connections.

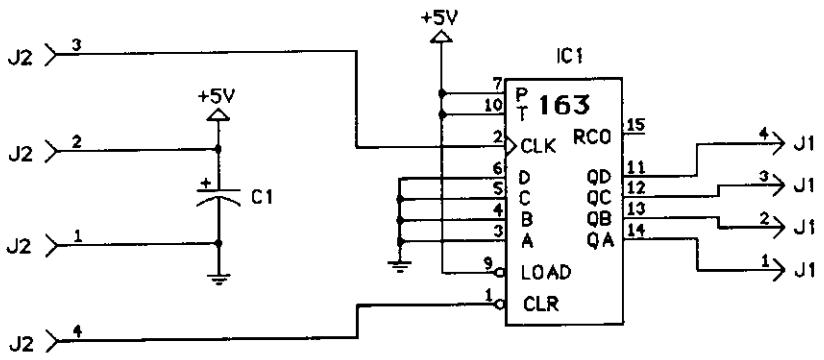
The pre-defined attribute fields "Power" and "Ground" are intended for this purpose.

- Power and ground pins can be added to each device symbol (using the DevEditor tool) and the connections shown as normal signal lines on the diagram. The standard DesignWorks libraries are not set up in this fashion.
- A separate symbol can be created for each device showing only power and ground connections. This allows all power wiring to be shown in one part of the diagram or on a separate page.

In fact, these methods are not mutually exclusive and can be used in combination as needed. The Report tool implements these methods by combining connections described in different ways into a single net in the netlist:

- Any number of special names can be specified as "signal sources" using the $SIGSOURCE keyword in the report form file. The Report Generator searches the attributes attached to each device for fields with these names. The pin number specified in the field will be attached to a list for the signal of the same name. E.g. the value "14" in the Power field will cause pin 14 on this device to be attached to the signal named as the power signal. Multiple pins can be specified using commas, e.g. "1,2,14".
- A Signal Connector device (such as the Ground and +5V symbols) can be used to create a logical connection between pins. See more information on Signal Connectors below.

These features will be illustrated by way of a simple example:



The attributes for the device IC1 were specified in the library entry for type "163" and are as follows:

```
Power     14
```

```
Ground    8
```

In the above diagram, the signal named "+5V", the symbols labeled "+5V" and the attribute entries "Power" will all be merged into a single list of connections in the netlist, as follows:

```
S00005    IC1-14    J1-1
S00007    IC1-2     J2-3
S00008    IC1-1     J2-4
S00009    IC1-11    J1-4
S00010    IC1-12    J1-3
S00011    IC1-13    J1-2
Ground    C1-2      IC1-3,4,5,6,8   J2-1
+5V       C1-1      IC1-10,7,9,16   J2-2
```

NOTE: The standard report form files add the entries in the Power field to a net named "+5V".

The demonstration files provided with DesignWorks provides more complex examples of this usage.

Power and Ground Naming Convention

The standard libraries provided with DesignWorks include the fields Power and Ground for the standard power and ground pins. Since field names cannot contain special characters, we recommend using the words "Plus" and "Minus" to create attribute fields for other supplies. For example, pins to be attached to a -12V supply would be listed in field "Minus12V". The Report Generator allows the name of the signal net to be different from the field name, if desired.

The supply and ground signal connectors can be found in the Connectors library. The standard device libraries include entries for Power and Ground and other standard power pins. As in the above example, other pins can be attached to these lines using symbols placed on the diagram.

See Chapter XII - Report Generation for instructions on adding other power nets.

Signal Connector (Power and Ground) Symbols

DesignWorks uses a type of pseudo-device symbol called a "signal connector device" to maintain connectivity between like-named power and ground symbols that are used on circuit diagrams.

As soon as a Ground symbol is placed on the diagram, the attached signal will be named "Ground" (the name will be invisible initially). This will cause it to be connected by name to any other signals having Ground symbols, or explicitly named "Ground".

Connectivity can be checked at any time by double-clicking on any ground or power line. This will highlight all other like-named lines on the diagram.

IMPORTANT: Signal connectors do not cause a logical connection to be made between hierarchy levels in a hierarchical design. These signals can be merged into a single net for netlisting purposes by specifying a $SIGSOURCE statement in the report form file. However, this processing is done as the report is generated and does not create a logical connection in the design, e.g. for interactive simulation.

See Chapter XII - Report Generation for more information.

Using Signal Connector Devices

Signal Connector Devices are placed on the diagram just like any other DesignWorks device. A set of standard power-supply symbols are included with DesignWorks in the Connectors library. If you connect two different signal connector devices together, you will be prompted to provide a name for the resulting signal.

Creating Signal Connectors in a Library

Signal Connector devices (like Page Connectors) are special primitive "pseudo-devices" in DesignWorks and can be created using the Set Primitive Type command in the DevEditor tool to select the SIGCONN primitive type.

NOTE: The signal attached to a signal connector device is actually named to match the pin name of the signal connector pin specified in DevEditor, not the type name. In most of the power and ground symbols provided with DesignWorks, these two names are the same. However it is possible to create a symbol called, for example, "Ground" in the library which actually names the attached signal "GND". The Ground symbol in the SPICE Devices library provided is an example of this in that it names the attached signal "0" to match the SPICE ground naming convention.

See Chapter X - Device Symbol Editing for more detailed information on this procedure.

Specifying Package Type Information

If the netlist output from DesignWorks is to be used to generate a printed circuit board, then it will be necessary to associate package type information with each device. This is done by specifying a field called "Package" with a keyword for each package type. Package type fields are included in the standard type libraries, but may be overridden if necessary by specifying a value for the "Package" field in the device attributes.

The keywords associated with the common package types are described in a technical note supplied with the package.

Connectors and Discretes

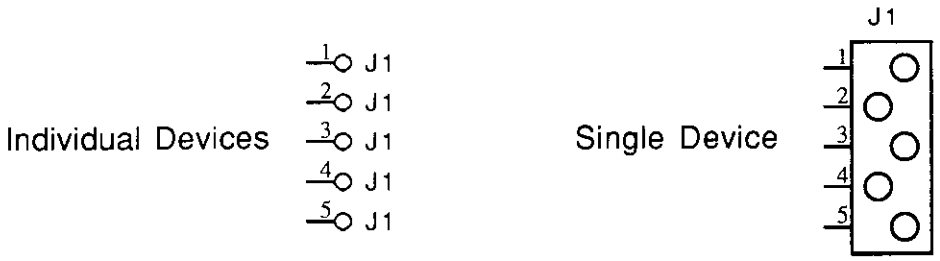
In DesignWorks, each symbol is considered to be a separate device and each device is normally assumed to be one IC package with a standard pin numbering scheme. Thus connectors and discrete components will require special consideration.

Handling Connectors

Connectors can be handled in one of two ways:

- A special symbol can be created for the connector with the appropriate number of pins and pin numbering specified for each pin. This can be done using the DevEditor tool to create a device symbol using your own picture.
- Each connector pin can be created as a separate, 1-pin device using the symbols from the Connectors library or a custom symbol. This option is preferable only if you need to spread the connector pins over different parts of the diagram. In this case, each "device" must be given the name of the connector and the pin number associated with that pin. We are relying on the fact that the report generator will normally merge all devices with the same name into a single component entry. Attributes (if any) are taken from the first device encountered with the given name.

Following is an example of these two methods:



NOTE: When the single-pin devices are used, every device must carry exactly the same name, although the names can be invisible if desired.

Handling Discrete Components

Discrete components such as capacitors, transistors, etc. can be handled just like any other device except for the following special considerations.

Pin Numbering on Discrete Components

Pin numbers are not normally placed on discrete component pins on a diagram. If pin numbers are omitted from a device, DesignWorks will normally put a question mark in the netlist item for that device. Two methods are available to provide pin numbers for netlisting purposes:

• To provide automatic numbering of discrete devices pins, the Report Generator provides the AUTONUMBER option, which can be specified in the form file. This option causes any device with less than or equal to the given number of pins to be numbered automatically if no pin numbers are present on the diagram. In the standard form file, this value is set to 3 to cover the common discrete components.

WARNING: This option assumes that the pin number order of the discrete components is not significant. If a specific order is important, do not use this method.

• Pin numbers can be assigned but left invisible. This is done using the Get Info command for either the pin or the device.

Package Types for Discrete Components

Discrete components do not normally have standard package types. If a package type is required for PCB layout purposes, then the "Package" field in the attributes must be set manually for each device.

Advanced Editing Procedures

The purpose of this section is to give you a quick outline of commonly-used schematic editing techniques in DesignWorks. Background on each of these techniques is provided earlier in this chapter and in other reference sections of the manual.

Adding a New Page to a Circuit

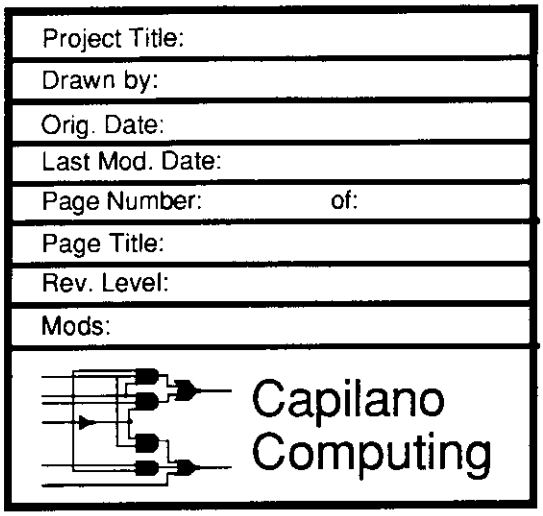
To add additional circuit pages to a circuit, select the Page command in the Drawing menu. The box that appears has controls allowing you to add, delete, re-order and add titles to pages. Clicking on the New Page button will add a new, empty page to the current circuit.

See the Page command in Chapter IX - Schematic Menu Reference.

Creating a Title Block

DesignWorks has a number of features which can assist in creating title blocks which will contain information that is automatically updated. A simple title block can be created simply using the random text feature by clicking the pencil cursor in the desired location on the diagram. After the text has been typed, click on the text block to select it, then use the Get Info command to set the text size appropriately and create a frame.

A more striking title block can be created using a graphic item imported from a Macintosh drawing program, such as the following:



First, paste the graphic on the diagram using the Paste command then select it and set it to be a background and border object using the Get Info command. Once it has been converted to background, it will not be selected again and you can use text notations to draw on top of it.

NOTE: Background objects can be selected by holding the ⌘ and Shift keys while clicking on them.

Text variables can be used to add dates and titles to the block. See the section on variables earlier in this chapter.

Creating a Custom Sheet Template

Creating a Fixed Sheet Size

A fixed sheet size is one that does not vary with the current Page Setup printer settings. In order to have a complete description of a sheet, you need to indicate to the program the following points:

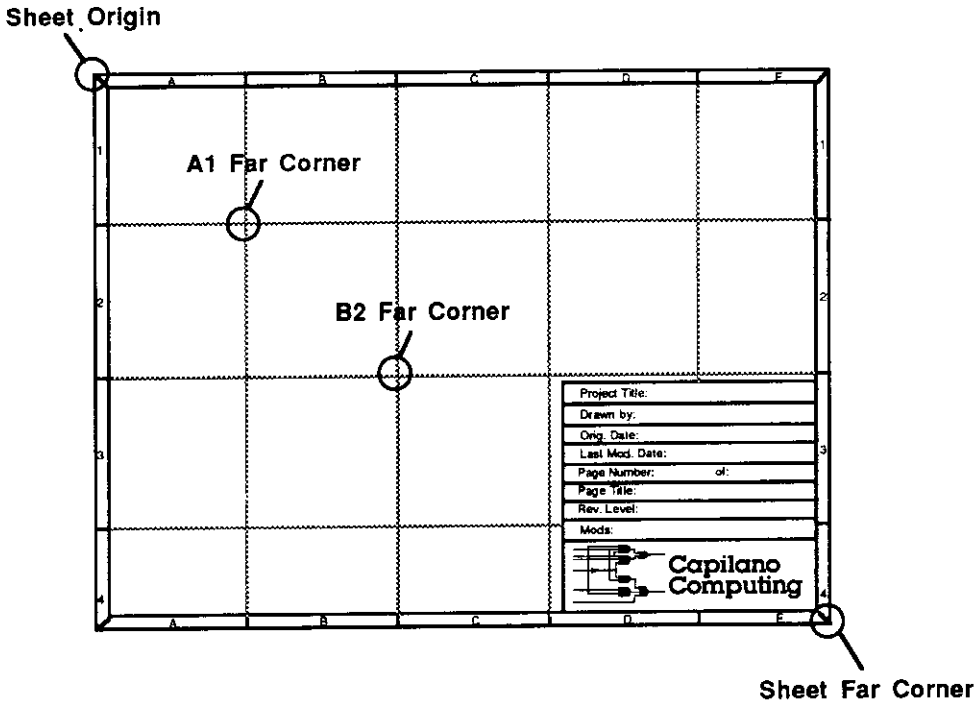
Point #1 - The sheet origin. This is the corner of the sheet where the A1 grid box is located. The origin point will be on the outside line of the border. Drawing will not be allowed outside this point.

Point #2 - The opposite corner of the A1 grid rectangle, i.e. where it meets the B2 grid rectangle. This point is used to determine the start of the regular drawing grid.

Point #3 - The opposite corner of the B2 grid rectangle. The space between Points 2 & 3 determines the grid spacing for the rest of the sheet.

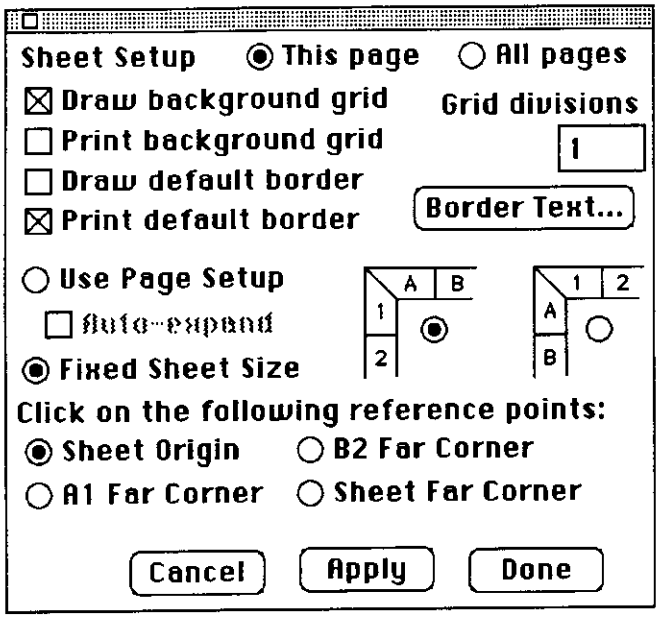
Point #4 - The far corner of the sheet. The point indicated will be the outside line of the border. Drawing will not be allowed outside this point.

In order to accurately click on these points in the circuit window, it is best to create a measured template picture using an external drafting application. This picture can be first pasted in the circuit at the desired location, then used as a reference for this procedure. Following is a sample A-size horizontal background picture with the reference points marked:



NOTE: The sheet origin can be in any corner. It is the corner that will be the origin (i.e. the "A1" box) for the reference grid.

Select the Custom Sheet Info command in the Drawing menu. The Sheet Info palette will appear as follows:



Click on the Fixed Sheet Size option. You will see the Sheet Origin button has become enabled. This indicates that you should scroll to the sheet origin position and click accurately on the origin point.

You will now notice that the A1 Far Corner button has been automatically selected. Click now at the opposite corner of the A1 grid rectangle.

Next, locate and click on the B2 far corner and sheet far corner points.

Select any other sheet option boxes desired, then click on the Done button. The sheet info palette will disappear and the displayed drawing area will be updated.

Creating Custom Sheet Border Graphics

Any picture or text objects can be made part of the sheet border. To do this:

- select the object.
- choose the Get Info command in the Options menu.
- enable the "Make Background" and "Make Border" switches.
- click OK.

The object will now be treated as part of the sheet background. To select it for deletion or editing, hold the Command and Option keys while clicking on it.

Setting Device Packaging Options

To set device packaging options, bring up the Device Info box by either one of the following methods:

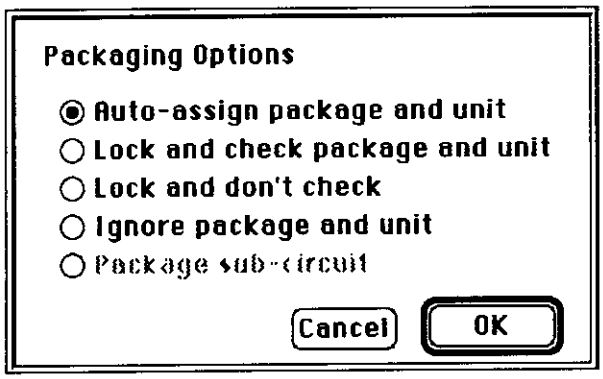
1) Select the device by clicking on it, then select the Get Info command in the Options menu.

OR

2) Hold the ⌘ key while clicking on the device, then select the Device Info command in the pop-up menu.

Next, click the Packaging Options button in the Device Info box:

This displays the following box:



Following is a quick summary of the meanings of these options. For complete information see Device Packaging earlier in this chapter.

Normal (Auto-assign package)	Name and unit number will be assigned automatically by the Packager.
Lock and Check	Name and unit will not be assigned, but the present values will not be assigned to any other device. The device will be checked for packaging errors.
Lock and Don't Check	The same as Lock and Check except that the package assignment is not checked for errors.
Ignore	Name and unit are not changed or checked and may be assigned to other devices.

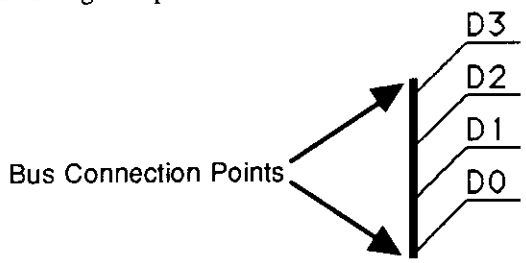
Package Sub-circuit	Ignore this device symbol and package devices in its internal circuit. This option will be disabled if the symbol has no internal circuit.

Bus Operations

Creating a Bus

A bus can be created by any one of the following methods:

- Select the Draw Bus tool (+) in the tool palette. Draw any desired contiguous set of lines on the diagram using the usual signal drawing techniques. This bus will have no internal signals initially. Signals will be added implicitly when it is connected to any breakout or bus pin.
- Create a breakout symbol using the New Breakout command (see below). The bus pin (backbone) of the breakout can now be extended using the normal pointer (↖) or the Draw Bus cursor. The bus will contain all signals specified in the breakout.



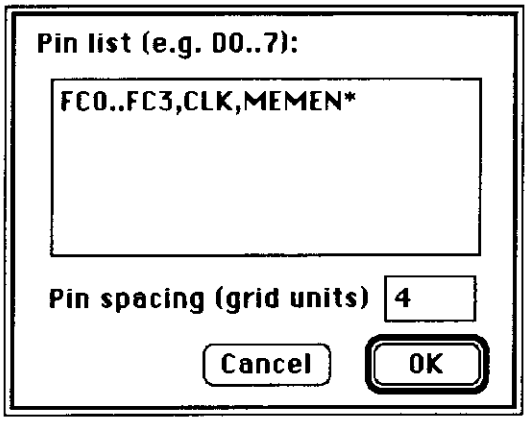
- Extend a line out from an existing bus pin on a device (see below) using the normal pointer or the Draw Bus cursor. The bus will contain all signals specified in the bus pin on the device. Connections between bus internal pins and bus internal signals can be changed using the Bus Pin Info command on the bus pin's pop-up menu.

Adding Signals to a Bus

There is no explicit command to add signals to a bus. Signals are added to a bus each time a breakout or device bus pin is connected to the bus. Any signals in the breakout or bus pin are implicitly added to the bus if they don't exist already.

Creating a Breakout

To create a breakout, select New Breakout command in the Options menu. If the new breakout is to be similar to an existing one, first select the similar breakout or the bus to which the new breakout is to be connected. The following dialog box will appear:



If a bus or breakout was selected on the circuit diagram then the breakout info box will display a list of the signals in that bus or breakout, otherwise it will be empty. If this list already matches the signals you want in the new breakout, then just click the "OK" button or hit Enter on the keyboard. Otherwise, edit the signal list, noting the following options:

- blanks or commas can be used to separate individual names in this list, therefore bussed signals cannot have names containing a blank or comma.
- a range of numbered signals can be specified using the following formats:

  `D0..7` or `D0..D7`

  is equivalent to

  `D0 D1 D2 D3 D4 D5 D6 D7`

  `D15..0`

  is equivalent to

  `D15 D14 D13 D12 D11 D10 D9 D8 ... D0`

```
D15..D00
```

is equivalent to

```
D15 D14 D13 D12 D11 D10 D09 D08 D07 ... D00
```

Note that the ".." format implies that bussed signal names cannot contain periods.

- the signals specified will always appear in the order given in this list from top to bottom in standard orientation. We recommend always specifying numbered signals from lowest numbered to highest, as in the first example above, since this matches the standard library symbols.
- there is no fixed limit on the number of signals in a bus, but we recommend dividing busses up by function (i.e. address, data, control, etc.) for ease of editing.
- any combination of randomly-named signals can be included in the list, as in the following examples:

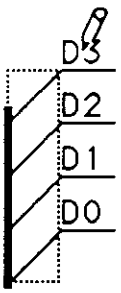
```
D0..15 AS* UDS* LDS*
CLK FC0..3 MEMOP BRQ0..2
```

Once the list has been entered, click on the OK button or hit the Enter key. A flickering image of the breakout will now follow your mouse movements and can be placed and connected just like any other type of device.

Editing Breakout Pins

The signal name notation that appears on a breakout pin is actually a pin attribute. It can therefore be edited by the usual attribute editing mechanisms, i.e. either:

- Select the pin and choose the Get Info command in the Options menu, then click the Attributes button, OR,
- Click the text cursor directly in the text on the schematic, as follows:



- Type the desired new name.

• Press the Enter key. The breakout pin and the attached signal will be renamed as entered.

IMPORTANT: The notation on the breakout pin is always the same as the name of the attached signal name. Changing the breakout pin renames the attached signal and will detach it from any like-named signals already in the bus.

Changing Bus Pin Connections

When a bus is connected to a bus pin on a device or hierarchical block, the bus internal pins will by default connect to signals with the same name in the bus. To change these default connections, use the Bus Pin Info command in the pin pop-up menu.

See Chapter IX - Menu Reference for more information.

Setting Text Styles

Pin Numbers

The text style for pin numbers is set globally for the entire design. It cannot be set individually for pins.

To set pin number text style:

1) Select the Design Preferences command in the Drawing menu.
2) Click on the Pin Text button.
3) Select the desired text font, size and style in the style box.
4) Click OK on the text box, then OK on the Design Preferences box.

Depending on the size of the design, there may be a short delay at this point while sizes and positions of text items are recalculated.

Attributes

The text style for attributes is set globally for the entire design. It cannot be set individually for each item or for each field.

To set attribute text style:

1) Select the Design Preferences command in the Drawing menu.
2) Click on the Attr Text button.
3) Select the desired text font, size and style in the style box.

4) Click OK on the text box, then OK on the Design Preferences box.

Depending on the size of the design, there may be a short delay at this point while sizes and positions of text items are recalculated.

Border Text

Text style can be set for the reference letters and numbers appearing in an auto-generated sheet border. To do this:

1) Select the Custom Sheet Info command in the Drawing menu.
2) Click on the Border Text button.
3) Select the desired text style.
4) Click OK on the text style box, then Done on the border info box.

Text Blocks

Text style can be set individually for each text block. To do this:

1) Select the text block by clicking on it. (If it is a background item, then hold the command and option keys while clicking on it.)
2) Select the Get Info command in the Options menu.
3) Click on the Text Style button.
4) Select the desired text style.
5) Click OK on the text style box, then OK on the text info box.

Setting Page Reference Format

To change the page reference format, select the Design Preferences command in the Drawing menu. Two aspects of the page reference format can be controlled:

- the "Format" text item controls the appearance of each page reference in the reference list.
- the "Max. Width" item controls the number of references that will appear on each line before creating a new line. For signals with many connections, this prevents unwieldy page reference strings.

Three characters are special in the format string:

P	will be replaced by the page number
X	will be replaced by the X grid position
Y	will be replaced by the Y grid position

The default format is "P-XY" which generates pages references like "2-C4" (i.e. page 2, grid position C4). All characters other than the special ones above will be placed in the page reference list verbatim.

The maximum length of a format string is 16 characters.

Setting an Alternate Name Prefix Field

The name prefix field is the field that is used to generate default names for devices as they are placed. The standard prefix field is "Name.Prefix". Two other fields are included in the standard DesignWorks libraries which can be used as prefix fields:

- "Name.Spice" is defined in the Discretes library and contains the prefix commonly used by SPICE-based analog simulators to identify part type.
- "Function" contains a short function code for each discrete and digital part. It can be used as an alternate prefix field to give descriptive names to parts as they are placed.

To set the prefix field:

1) Select the Set Design Attributes command in the Options menu.
2) Select the PrefixField item in the field list.
3) Type the name of the desired prefix field in the data box.
IMPORTANT: The name must match an existing field name exactly, including case, or it will be ignored.

Chapter VI - Attributes

DesignWorks allows arbitrary blocks of text to be associated with any device, signal or pin in a design, or with the design itself. The blocks of text are called *attributes*. Attributes have a wide variety of uses, including:

- displaying device name, package assignment, part type, value, etc.
- storing manufacturing information such as catalog numbers, manufacturer's name, price, part description, etc.
- storing data for use by external systems such as simulators, PCB layout, analysis.tools, etc.

In addition, many DesignWorks features use attributes as temporary or permanent storage locations for data relating to a device or signal. For example, each device in a design has the name and location of the library it was read from stored in an attribute field. This allows the library to be located easily for later updating.

Attribute Organization

Attribute Definition Table

Attributes are divided into named *fields*. The list of allowable field names is stored with each design in the *attribute definition table*. For each field, the table contains:

- the field name (up to 16 characters)
- what types of objects that field can be used in (devices, signals, pins or design)
- data entry constraints (maximum length, carriage returns, etc.)
- default settings, such as whether the field should be made visible on the diagram.

Once a field is defined in the table, its name is never typed again. When attribute data is entered for a selected object, the field name is picked from a list of allowable fields. This eliminates any chance of accidentally misspelling the field name in one object.

Predefined Fields

When a new design is created, a default attribute definition table is created with the standard, predefined fields. Predefined fields cannot be deleted from the design since many of them have specific internal functions in DesignWorks.

See Appendix C - Predefined Attribute Fields for a complete listing of predefined fields.

User-defined Fields

You can add more fields to a design at any time using the Define Attribute Fields command in the Options menu.

Primary vs. Secondary Fields

DesignWorks has 40 or more predefined fields that are used for various program features and more can be added by the user. Most of these fields never need to be viewed or set directly by the user. To reduce clutter, attribute fields can be defined as being *primary* or *secondary*. When entering attribute data, you can choose whether you wish to see "primary fields only" or "all fields". The Attribute Probe tool (?) in the tool palette shows only primary fields.

The primary/secondary setting is for convenience only and has no effect on the meaning or internal usage of the field. This setting can be changed at any time using the Define Attribute Fields command in the Options menu.

Definition vs. Instance Fields

In a hierarchical design, a single device, pin or signal in a sub-circuit may actually represent several physical objects. This happens if the parent device containing the sub-circuit is used more than once in the design.

For this reason a distinction is made between fields that are associated with the definition of a sub-circuit and fields associated with the instance.

- Fields associated with the definition will be the same in all instances of the sub-circuit. Changing the value in one instance will affect all instances equally.

• Fields associated with the instance can take on a different value in each physical instance of the device, pin or signal. Changing the value in one instance has no effect on the others.

When defining a new attribute field for use in a hierarchical design, careful consideration should be given to whether to consider it a definition or instance field. If the value of the field has a significant impact on the logical functioning of the sub-circuit, then the field should probably be a definition field. Thus, if two sub-circuits require different values in this field, then separate definition circuits will have to be created. This makes it clearer to other users of the design that a distinction exists.

See Chapter VII - Hierarchical Design for more information.

Name and InstName Fields

In DesignWorks, the single most important attribute field is the "Name" field. It can be associated with devices or signals and is used to uniquely identify an object within its circuit. The Name is a *definition* field.

Note that the Name field is unique within a circuit, not necessarily within the entire design. For this reason, the InstName field is used in hierarchical designs. InstName is an *instance* field, meaning that it can have a different value for every physical device in a hierarchical design. See the section "Names in Hierarchical Designs" below. The InstName field is not normally used in Flat mode designs.

Names may contain any letters, numbers, "_" (underscore) or "." (period) characters. They are restricted in length to at most 15 characters. The name associated with an object can be placed anywhere on the diagram and will be removed if the object is removed.

Invisible Names

Like other attribute fields, the Name field can be associated with a device or signal without displaying it on the diagram. The visibility of the name can be set using the Get Info or Attributes command associated with the device or signal in question.

NOTE: When signals are connected by name, only visible names affect connectivity. Two signals with the same invisible name will not be connected. See Chapter IV - Schematic Editing for more details.

Device Names

Typical device names might be "U23", "C4", "XTAL1", etc. The Name field for devices is equivalent to the "reference designator" used in other systems.

The Name is distinct from the "type name" that is used to distinguish the type definition that is read from a device library. The "type name" is not an attribute, but a fixed name associated with a device symbol when it is saved in a library. Typical type names are "74LS138", "MC68000L8", "SPDT Switch", etc.

As well as appearing as a text notation on the diagram, the device Name field is used by the following DesignWorks functions:

- if the design is in flat hierarchy mode, then the Packager tool places the assigned package name in the Name field. In other modes, the "InstName" field is used.
- the device name is used in Report Generator output, such as netlists and bill of materials reports.
- the device can be located by name using the Find tool.
- the device name associated with a Page or Port Connector is used to make logical connections.

Signal Names

Typical signal names are "CLOCK", "ADDR12", etc. Signal name is referenced by the following DesignWorks functions:

- the signal name is used in Report Generator output, such as netlists.
- the signal can be located by name using the Find tool.
- signals can be logically interconnected by name.

See more information on signal names and logical connections in Chapter IV - Schematic Editing.

Names in Hierarchical Designs

In a hierarchical design, a single device in a sub-circuit may actually represent several physical devices. This happens if the parent device containing the sub-circuit is used more than once in the design.

See Chapter VII - Hierarchical Design for an example of this situation.

For this reason another attribute field is needed that will be unique for each and every physical instance of a device in the design. The "InstName" field serves this purpose.

If the design is in "physical" hierarchy mode, the InstName field is used by the Packager to store the package assignment. If the Packager is not being used, the InstName field can be used to store notations about the physical placement of the device.

NOTE: In a hierarchical design, the Name field can be thought of as a "logical" name and the InstName field as a "physical" name. The "logical" name should be assigned by the designer based on the device's function in the circuit, not on its physical location.

The visibility of the Name and InstName fields on the diagram can be set separately. If may be desirable to only show the Name field on the diagram during the early stages of design development. The physical package assignments can be shown later when needed for manufacturing.

Value List Fields

For attribute fields that can only take on one of a known list of values a *value list field* can be created. This mechanism is used to create the "Unit" and "Part" fields used in all standard DesignWorks libraries. These standard fields can be used as an example.

A value list field has the same name as the associated field but with ".List" appended. When a field named "xxx.List" is seen by the program, it does the following:

- when the attribute edit box is displayed for a given object and the field "xxx" is selected, a pop-up menu is displayed containing all the possible values for the field. The user may then select the appropriate value from the menu.
- when a pop-up menu is displayed by the user using the key, a sub-menu with the field name "xxx" as its title will appear. The list of possible field values will be displayed in the sub-menu. The user can select one of these items to directly change the value of field "xxx".

NOTE: DesignWorks does not prevent the user from entering a field value that is not in the list.

Value List Data Format

The value list field must contain a sequence of textual values following these rules:

- items must be separated by commas. A comma will always be taken as a separator and therefore cannot be part of the value.
- the value string can contain any characters other than commas, but leading and trailing blanks will be removed.
- the length of a value string is limited to 16 characters.
- the maximum number of values is limited to 256.

Normally, a default value for the "xxx.List" field should be defined with the symbol in the library using the DevEditor tool. In this way, the value list will always be available whenever this device type is selected from the library. If the field is defined in only one device on the diagram, the value list will only be available for that device.

Default Position Fields

For attribute fields that are to be displayed on the schematic, an associated field called a "default position field" can be created. This mechanism is used to provide default positions for the "Name" and "Part" fields used in all standard DesignWorks libraries. These standard fields can be used as an example.

A default position field has the same name as the associated field but with ".Pt" (for Point) appended. When a field named "xxx.Pt" is seen by the program, it uses the values in this field to generate a default position for the "xxx" field.

Default Position Data Format

The default position field must contain a sequence of textual values following these rules:

- two items, an X and a Y position, must be specified, separated by a comma. Leading an trailing non-numeric characters are ignored.
- the X and Y values are each decimal numbers representing an offset from the top-left corner of the device measured in 1/1000". Negative values are allowed (negative is up and left).
- the X coordinate can be optionally followed by a letter indicating horizontal justification, as follows: "L" for left, "M" for middle, "R" for right. I.e. the given number specifies the offset to the left, middle

or right position of the displayed text. If no letter appears, middle is assumed, i.e. the text will be centered at the given position.

• the Y coordinate can be optionally followed by a letter indicating vertical justification, as follows: "T" for top, "M" for middle, "B" for bottom. I.e. the given number specifies the offset to the top, middle or bottom position of the displayed text. If no letter appears, middle is assumed, i.e. the text will be centered at the given position.

Some examples of value points are:

0R,0T	The value will extend right and down from the top-left corner of the device.
150,-75	The value will be centered at a position above and to the right of the top-left corner.
(400L,100)	The value will extend left from a point to the right of the top-left corner of the device. (Leading and trailing characters are ignored.)

Normally, a default value for the "xxx.Pt" field should be defined with the symbol in the library using the DevEditor tool. In this way, the value will be available for setting the default position of the associated field when the device is placed on the schematic.

Default Values

A device symbol can have associated with it predefined default values for any number of fields. Values can be specified for the device itself, and independently for each pin on the device.

When the standard attribute data box is displayed for a device, you will see a button labeled "Use Default Value". If this button is grayed out (disabled), then there is no default value or the value shown is already the default.

See Chapter X - Device Symbol Editing for more information on creating default attribute values.

Attribute Field Options

When a new field is created using the Define Attribute Fields command a number of options are available which determine how that field will behave in your design. These options are described in the following sections.

IMPORTANT: Changing any of the following options in a field that has already been used in the design has no effect on items already displayed on the schematic.

TIP: To make a temporary change (e.g. to display a field that is not normally displayed) you can use the Duplicate button in the Define Attribute Fields box to create a new field containing the same data but with new display options. This field can be deleted again when no longer needed.

Keep with Instance

This option applies only in Physical Hierarchy mode. When this switch is checked, the selected attribute field may take on a different value for each instance of a device, pin or signal.

See Chapter VII - Hierarchical Design for more information on instance data.

Group with Name

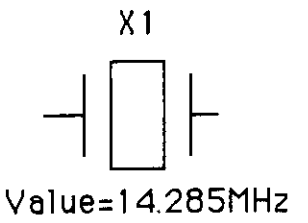
When this switch is checked, any value entered for this field will be displayed beneath the Name field by default. When the Name is moved, this field will also be moved.

This option can be used for fields that are typically shown with the name on a schematic. E.g. the Unit field (gate unit) has this setting enabled by default.

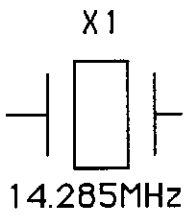
Show Field Name

When this switch is checked, the field name will be shown with the value on the schematic.

With Show Field Name



Without Show Field Name



The Show Field Name option can also be controlled for each individual item on the schematic using the Show Field Name command in the attribute pop-up menu.

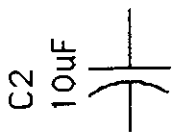
Allow Carriage Returns

If this switch is checked, then the Return key may be used to enter Carriage Return characters into this field.

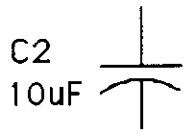
Rotate Text With Object

If this box is checked, displayed values will be rotated if the associated object is rotated on the schematic. Otherwise, when the object is rotated, the text will move to an appropriate new position but be drawn in the normal orientation.

Rotate With Object



No Rotate With Object



Visible by Default

If this box is checked, whenever a value is entered for this field it will be displayed on the schematic in a default position. If the field is found in the default attributes for a device, the field will be displayed automatically on the diagram when the device is placed.

NOTE: Changing this setting has no effect on values already in the circuit. See the note at the front of this section.

In Primary List

If this box is checked, the field will be considered a "primary" field and will be displayed in all attribute entry boxes when "Primary Only" is selected. It will also appear in the window that is displayed by the Attribute Probe (?) cursor.

Allowed in Object Types

This group of check boxes allows you to select which types of schematic objects may have this field associated with them. You can check any one or more boxes.

Maximum Length

You can specify a maximum length (in characters) for the field data.

Attribute Limitations

Attribute fields have the following specific limitations:

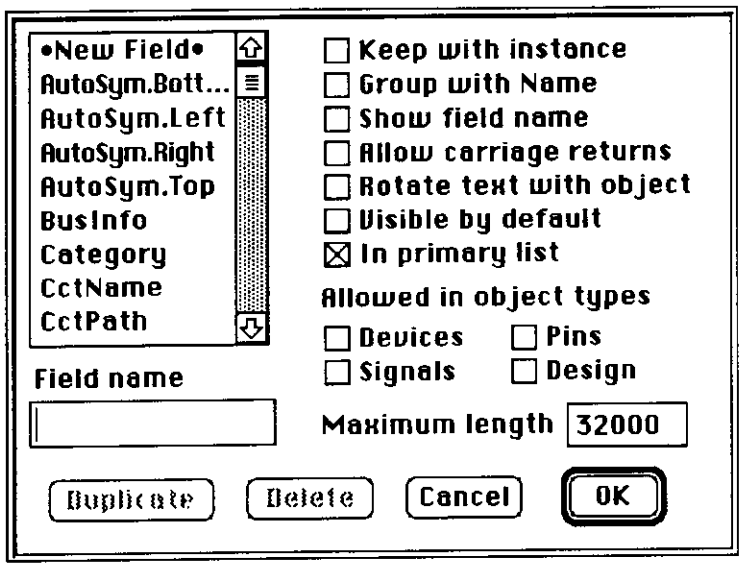
- Maximum length of field name: 15 characters
- Maximum length of field data item: 32,000 characters. Note: In principle, binary data could be pasted into an attribute field although this is not recommended since DesignWorks provides no mechanism for displaying or editing this data.
- Maximum number of user-defined fields: 900
- Maximum number of displayed positions of a single attribute item: 100

Like all other circuit data, the amount of attribute data that can be associated with a design is limited by available memory.

Attribute Procedures

Defining a New Field

All changes in field definitions are done using the Define Attribute Fields command in the Options menu. Selecting this command displays the following dialog box:



To create a new field:

1) Select the •New Field• item in the list, if it is not already.

2) Type the name of the field in the Field Name box.

3) Select one or more of the boxes under "Allowed in object types".

4) Optionally enter a maximum length.

5) Optionally enable any of the other field options. NOTE: These options affect only new data as it is entered, not values already in the design.

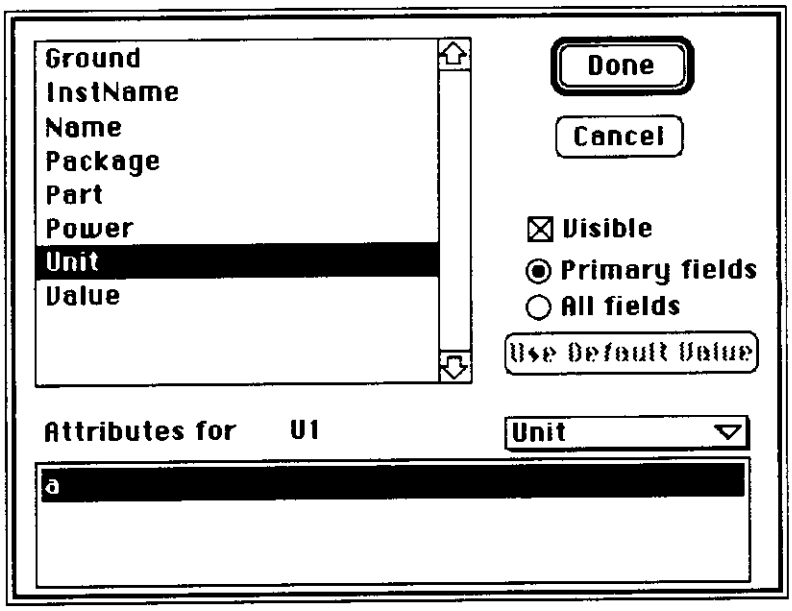
6) Click OK. The new field will now appear in the attribute data entry box for the selected object types.

See the description of the Define Attribute Fields command in Chapter IX - Menu Reference for details on the field options.

Editing Attribute Data (General)

NOTE: The same attribute edit box is used to enter data for all object types. This section discusses the general operation of this box. The following sections discuss each object type.

The following edit box is used to enter or edit attribute data:



Basic Procedure

To edit the contents of a field, simply select the field name in the list. The current contents of the field will be displayed in the editable text box. Edit this value using the usual Macintosh text editing techniques. Press the OK button or select another field when done. If the data you typed exceeded the maximum length for the field, or if it contained invalid characters for the field, then you will be asked to correct the data.

You can view or edit as many fields as desired while in this box. No changes are made to the actual design data until you click the OK button. Clicking Cancel will abandon all changes made while in this box.

Default Value

Clicking the Use Default Value button sets the value for the selected field to the default value stored with the symbol. If this button is inactive (grayed out) then the value is already the default value, or no default value is present. Only devices and pins can have default values.

Value List Pop-up Menu

If the selected field has an associated value list field, then a pop-up value menu will appear just below the "Use Default Value" button. An item selected out of this menu will cause the value text box to be updated accordingly. This does not prevent the text data from being edited directly.

Editing Device Attribute Data

The Browser tool provides a convenient alternate method of editing attributes in many devices or signals at once. See the Chapter XII - Browser Tool.

The attribute edit box can be entered in one of two ways:

- click on the device to select it, then select the Get Info command in the Options menu, then click on the Attributes... button.
- pop up the device menu by holding the ⌘ key while clicking on the device. Select the Attributes command.

The standard attribute edit box will appear.

Select the desired field by clicking on it in the list. The current contents of

Displaying an Attribute on the Schematic

To display attribute text on the schematic:

1) Hold the ⌘ key while clicking on the device, signal or pin you wish to attach the attribute to. (Design attributes cannot be displayed on the schematic.) This will display the pop-up menu.

2) Select the Attributes command. This will display the attribute data box.

3) Select the desired field by clicking on its name in the field list. If the desired field is not a Primary field, you will have to click on the "All Fields" button in order to see it in the list.

4) Edit the attribute value as desired.

5) Turn on the Visible switch.

6) Click OK.

The attribute text will now be displayed in a default position near the device or signal. It can be dragged to the desired location using the pointer cursor.

Rotating Attribute Text

To rotate an attribute text item that is already displayed on the schematic:

1) Hold the ⌘ key while clicking and holding on the text item. This will display the attribute pop-up menu.

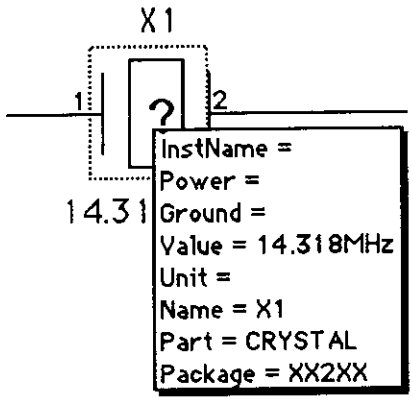
2) Select the Rotate Right or Rotate Left command.

Probing Attributes on the Schematic

The contents of all primary fields in an object can be viewed with a single click using the Attribute Probe tool.

1) Select the Attribute Probe command in the Edit menu or click on the ? tool in the tool palette.

2) Click and hold on the device, signal or pin in question. The dot at the base of the question mark is the "hot spot" that determines which item is selected.



Setting Attribute Text Style

Attribute text style is set globally for the entire design. There is no way to set text style for an individual attribute item.

WARNING: Changing the attribute text style affects all visible attributes throughout the design. This may alter text alignment and position to accommodate a different text size.

To set the global text style:

- choose the Design Preferences command in the Drawing menu.
- click on the "Attr Text..." button.
- select the desired font, size and style, then click OK.
- click OK on the Design Preferences box.

Depending on the size of the design, there may be some delay at this point. The program must check all visible attribute items to see if there position and framing is affected by the text change.

Creating a Value List Field

The creation of value list fields will be illustrated by way of an example.

Suppose you wish to create a symbol for a device which will be simulated using an external simulator. This symbol can be associated with one of three models which have the same pinouts, but different timing parameters. The model name is to be included in the netlist.

1) Using the Define Attribute Fields command, create two new fields, called "Model" and "Model.List". For both fields, check the "Devices" box indicating that these will be associated with devices.

2) Using the DevEditor tool, create the desired device symbol, or open an existing one for editing. Using the DevEditor's Part Attributes command, create a field called "Model.List" containing the desired list of values. For example:

```
MODEL1, MODEL2, MODEL3
```

3) Save the edited symbol to a library. Select the part from the library in the usual way and place it on a circuit diagram.

4 Hold the ⌘ key and click on the device. You will now see at the bottom of the pop-up menu a new item called "Model". Moving down to this item will display a new sub-menu containing the values entered in the "Model.List" field.

This page is intentionally left blank.

Chapter VII - Hierarchical Design

This chapter provides background and detailed procedures for the hierarchical design features of DesignWorks.

General Concepts

What is Hierarchy?

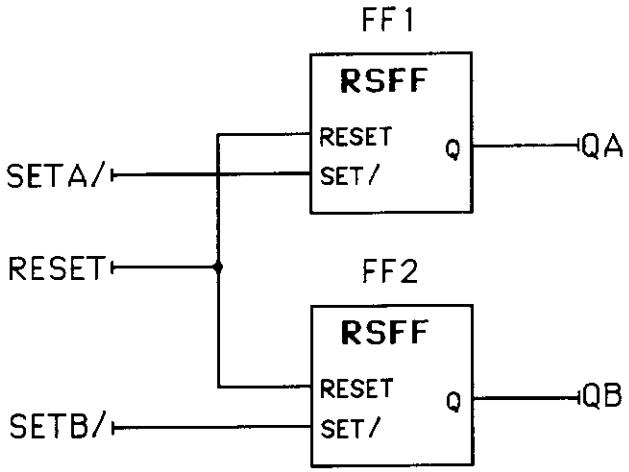
"Hierarchy" refers to the ability to have a "device" symbol in a schematic actually represent an arbitrary circuit block. The "pins" on the device symbol represent connections to specific input-output points on the internal circuit. For clarity, a device symbol that represents an internal circuit will be called "hierarchical block".

Hierarchical design provides a powerful way of representing complex designs in compact and readable form. A top-level diagram of your system can show only major functional blocks. These blocks can then be opened to show more and more design detail.

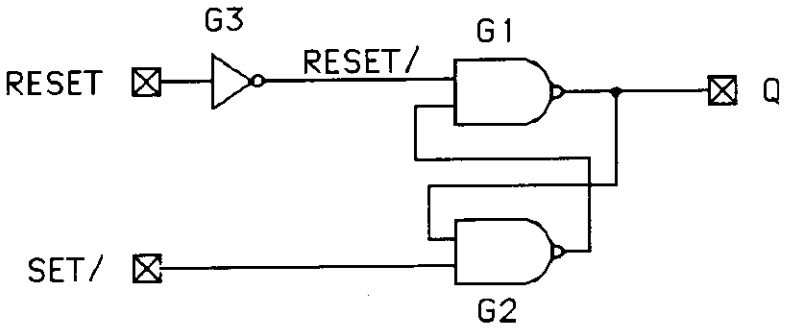
Hierarchical design in effect adds a "third dimension" to a schematic diagram. It also raises some complex issues that should be understood before embarking on a major design. Please review this chapter carefully before making extensive use of the hierarchical features of DesignWorks.

A Simple Hierarchy Example

The following diagram is the master circuit, or top level, of our design example:



Note that it contains two symbols, both representing hierarchical blocks. Both symbols are of the same type, "RSFF", and therefore share the same internal circuit definition. The two blocks are named "FF1" and "FF2". Opening either one of these blocks reveals the following internal circuit:



This circuit consists of three device symbols, "G1", "G2", and "G3", representing physical devices, and a number of "port connector" symbols. The port connectors define the interface between the internal circuit and the pins on the symbol representing it.

Note the following characteristics of this simple design:

- The block "RSFF" has been used twice, so there are actually two G1s, one inside FF1 and one inside FF2. We say that there are two *instances* of G1. Similarly for G2 and G3.
- The signals "SET/", "RESET" and "Q" in the internal circuit will actually get absorbed into the attached signals in the parent circuit because they are attached to port connectors. They do not exist independently in the physical circuit.

• Signal "RESET/" does not connect to a port connector, so it represents a separate signal in the internal circuit. Like the devices G1, etc., each signal in RSFF actually represents two physical signals.

These characteristics of a hierarchical design raise the following issues:

• Device and signal names inside hierarchical blocks are not unique if the block has been used more than once. Therefore, a mechanism has to be found for generating a unique name for use in a netlist or bill of materials.

• Signals attached to port connectors get absorbed into the signal attached to the parent symbol's pin. If this too is attached to a port connector, the process repeats upward until a unique signal is found. The name that will appear in the netlist will be the name of the highest-level signal.

• A single symbol in the definition of an internal circuit can actually represent two or more physical devices. A mechanism must be provided to store separate information for each of these physical devices, e.g. for PCB placement or layout information.

• When transferring data to PCB or simulation systems, it is usually necessary to produce a "flattened" netlist. I.e. a netlist representing the same circuit as if it had been expanded out on a single circuit level.

These issues are addressed in the following sections.

Definition vs. Instance

In DesignWorks, circuit data for hierarchical blocks is separated into two groups:

• Definition data is all information about the structure and connectivity of a circuit, plus the contents of all attributes not marked as "instance" fields. Definition data is associated with the parent symbol and will therefore be identical for all usages of the parent symbol.

• Instance data refers to data kept with each physical instance of a device. For example, in the simple design used in the previous section, gate G1 will have two sets of instance data associated with it, one for each physical device that it represents. Instance data consists of pin numbers, interactive simulation data and any attribute fields marked as "instance" fields.

In a hierarchical design, only one set of definition data is kept for each hierarchical block type, whereas separate instance data is kept for each instance.

Hierarchy Mode

DesignWorks 3.0 implements a true hierarchical design structure. This means that "device" symbols appearing in a circuit can actually represent another, nested circuit. This sub-circuit can be opened and edited at any time in separate circuit window. The term "design" is used in this manual to refer to a complete logical system, including the top level circuit and all sub-circuits it contains.

DesignWorks 3.0 has three hierarchy modes to address different design situations.

Flat Hierarchy Mode

"Flat" Mode is the simplest mode and is intended for small designs or those imported from older versions of DesignWorks. Despite the name, it is still possible to create and edit internal circuits, but they are assumed to be for simulation or analysis only. Devices in sub-circuits will not be assigned to physical packages by the Packager tool and will not appear in any netlist output from the Report tool.

Physical Hierarchy Mode

Physical mode allows full use of hierarchical design, while still allowing data associated with individual device instances to be stored at any point in a design. It is intended for larger board-level or FPGA designs where interactive simulation is used or PCB layout information is to be kept with the design.

Pure Hierarchy Mode

Pure mode stores only definition data for each device in a design. The "Keep With Instance" setting in attribute fields will be ignored. This allows very large designs to be represented in a small amount of memory, but does not allow interactive simulation or physical layout data to be associated with the design. It is intended for VLSI or other large-scale design applications.

The choice of hierarchy mode is not cast in stone once made since you can switch modes at any time. "Flat" and "Physical" modes are very similar and switching between them will normally only require that gate packaging be redone. "Pure" mode is substantially different, however, and should only be used after reading the discussion later in this manual.

WARNING: Switching to Pure hierarchy mode from any other mode will cause all instance data to be lost. THIS CANNOT BE UNDONE!!! See the following section.

Effect of Changing Hierarchy Mode

The following table summarizes the effects of switching between hierarchy modes.

Flat -> Physical	• No data is lost • Attribute field usage for device packaging changes. Packaging will have to be redone
Flat -> Pure	• Any instance data associated with internal circuits will be lost
Physical -> Flat	• No data is lost • Attribute field usage for device packaging changes. Packaging will have to be redone
Physical -> Pure	• Any instance data associated with internal circuits will be lost
Pure -> Flat	• No data is lost
Pure -> Physical	• No data is lost

Netlists from Hierarchical Designs

When producing a netlist for use by an external system, it is necessary to determine whether a "hierarchical" or "flattened" netlist is required.

Hierarchical Netlists

A hierarchical netlist retains the hierarchical structure of the original design. It contains only a single description of each type of hierarchical block used in the design. Normally, the lowest-level blocks are written out first, followed by higher blocks that refer to them, followed finally by the master circuit for the design. By its nature, a hierarchical netlist can contain only definition data.

This format is used by many simulators and FPGA packages, e.g. SPICE.

Generally, any package that will accept a hierarchical netlist can also accept a flattened one. It may be useful to use a flattened netlist in cases where unique instance data was stored with devices or signals in internal circuits.

Flattened Netlists

A flattened netlist has had all information about hierarchical structure removed from it. Every device and signal instance is listed separately with its instance data. Flattened netlists are normally required for PCB formats since name, unit and pin number assignments must be different for each device instance.

See more information on hierarchical netlists in Chapter XII - Report Generation.

Hierarchical Names

When producing a flattened netlist from a hierarchical design, it is usually necessary to generate a unique name for each device and signal in the design. This can be done in one of two ways:

• Manually or automatically assign a unique identifier to each item and store it in an "instance" attribute field. For example, in physical hierarchy mode, the Packager assigns a unique name to each device instance and stores it in the "InstName" field.

• Generate a unique identifier consisting of the definition name (i.e. "Name" attribute field) of the device or signal, prefixed by the names of all parent devices. For example, in the simple design example given at earlier in this chapter, the six physical devices would be called:

```
FF1/G1
FF1/G2
FF1/G3
FF2/G1
FF2/G2
FF2/G3
```

Assuming the Name field is unique within each circuit level, this is guaranteed to produce a unique identifier for each instance.

The "/" slash character is the standard name separator used in DesignWorks but can be changed by the user if needed. Refer to the Procedures section of this chapter.

Hierarchical Block Primitive Type

Hierarchical block symbols are simply device symbols which have the primitive type "SUBCCT", or sub-circuit. Device symbols with any other

primitive type cannot be used as hierarchical blocks. SUBCCT is the default primitive type when creating symbols with DevEditor, so it is normally not necessary to change this setting.

Printing Hierarchical Designs

A hierarchical design has an extra dimension which must be taken into account in determining the order of page printing.

Print Page Order

When a design is printed, the master circuit of the design is printed first in its entirety. Then, each hierarchical block is printed, followed by all hierarchical blocks it contains. For example, given the following structure of nested hierarchical blocks:

The print order would be: A, L, W, X, M, Y, Z. The order of printing of hierarchical blocks inside a single circuit is determined only by internal storage order and cannot be controlled by the user. E.g. W and X could be reversed above.

Scope of Printing

The Print Range command in the File menu determines how much of a design is printed using the Print command.

See more information in Chapter IX - Schematic Menu Reference.

Page Numbering

When a hierarchical design is printed, the $PRINTPAGENUMBER text variable can be used to apply sequential page numbers to the printed sheets.

NOTE: This text variable only applies during a Print operation. When it is drawn on the screen, it is interpreted the same way as the PAGENUMBER variable, i.e. it gives the page number within the circuit.

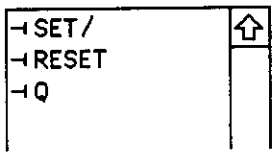
See more information on text variables in Chapter V - Advanced Schematic Editing.

Port Interface

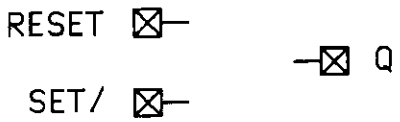
Signal connections between the levels of a hierarchical design are made using port connector symbols. With the exception of power and ground nets, all connections between levels must pass through a port connector.

Port/Pin Naming

The relationship between the port connector in the sub-circuit and the pin on the parent device symbol is established by matching the pin name on the parent device with the Name of the port connector. E.g. if we were to open the RSFF device used in the example above using the DevEditor tool, we would see the following pins listed:



For a complete port interface, a port connector must exist in the internal circuit named to match each one of these pins. In this case, the following port connectors would be required (ignoring all other internal circuitry):



The port interface is rechecked whenever any change is made. Thus, as soon as a port connector is added or removed, or its name changed, the port interface has been updated to reflect the new logical connections. However, to avoid excessive warning messages, error checking is performed only when an internal circuit is opened or closed. A warning box will be displayed if any error is found. This checking cannot be disabled.

NOTE: The name of the port connector's pin and the name of the signal attached to the port connector are not significant in making the port association. Only the contents of the port connector's Name field is used. Note the different rules for bus ports below.

Port Pin Type

If pin type (i.e. input/output/bidirectional, etc.) is important for your design, then the following rules should be noted when creating and using Port Connectors:

- The pin on the Port Connector symbol is normally of the opposite type to the corresponding pin on the parent device symbol. E.g. A signal coming in to the hierarchical block is actually an output from the port connector pin.
- DesignWorks does not check correctness of pin types on port connectors.
- When an internal circuit is auto-generated, port connectors of a type opposite the parent pin are placed automatically.

Bus Ports

Connections can be made between busses across hierarchy levels using Bus Port Connectors. Bus pins on a parent device symbol must be matched with a Bus Port Connector having identical internal pins. For this reason, Bus Port Connectors must always be custom made, either using the auto-generate function described below, or using the DevEditor tool.

Bus Pin Name Matching

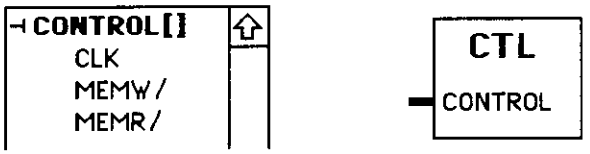
Note the following rules for name matching in bus ports:

- As with other Port Connectors, a Bus Port Connector must be given a Name exactly matching the pin name of the bus pin on the parent device.
- The internal pins in the parent bus pin must exactly match the internal pins on the Bus Port Connectors bus pin.
- The pin name of the bus pin itself on the Bus Port Connector is not significant.
- As with normal ports, the names of the signals attached to the Bus Port Connector's pin are not significant.

Bus Pin Example

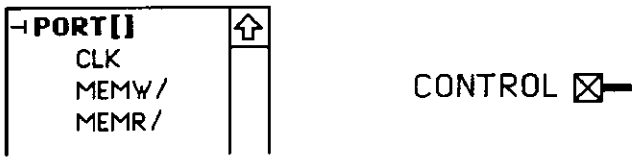
For example, the following simple device has a bus pin called CONTROL containing internal pins CLK, MEMW/, and MEMR/.

Pin List (in DevEditor) Hierarchical Block Symbol



The corresponding Bus Port Connector to be used inside this device would look as follows:

Pin List (in DevEditor) Port Connector Symbol



The comments above under "Port Pin Type" apply to each internal pin in a bus pin. Remember that the name of the bus pin in the port connector is not significant.

Power and Ground Connections

Power and ground symbols (i.e. Signal Connector devices) do not make an immediate logical connection across hierarchy levels. For this reason, signal connectors should not be used to make active signal connections for interactive simulation purposes.

However, it is possible to merge power and ground nets across levels in netlist output using the Report Generator's SIGSOURCE function.

See Chapter XII - Report Generation for more information.

Auto-Generated Internal Circuit Template

If an empty hierarchical block symbol is opened (either by double-clicking on it or using the Push Into command), Design Works will create default "template" for the internal circuit using the following rules:

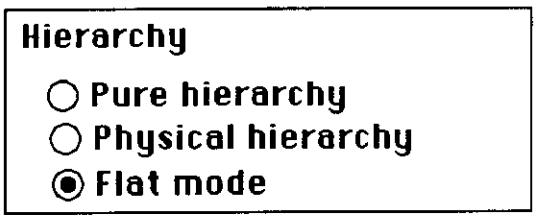
• A default port connector symbol is placed for each pin on the parent block, placed according to their position on the symbol.

• Each pin on each port connector has its type set according to the rules outlines earlier in this chapter.

Hierarchy Procedures

Setting the Hierarchy Mode

To set the hierarchy mode for a design, select the Design Preferences command in the Drawing menu. The following controls allow selection of hierarchy mode:



Click on the desired button to select a new hierarchy mode, then click on the OK button. A warning box will ask you to confirm the mode change.

WARNING: See the reference section earlier in the chapter for warnings about potential loss of data when switching hierarchy mode.

Creating a Hierarchical Block - Top Down

To create a hierarchical block from the top down, the following steps must be followed:

1) Create the block symbol using DevEditor.
2) Place the symbol on the schematic.
3) Create the internal circuit by pushing into the new symbol.

These steps are described in the following sections.

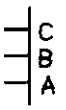
Creating a Block Symbol

Symbols for hierarchical blocks are created using the DevEditor tool. DevEditor provides numerous tools and functions for symbol editing. We will provide only a quick summary of one technique.

1) Invoke the DevEditor by selecting its name in the Tools menu.

2) Select the Auto Create Symbol command in the DevEditor menu.

3) In the "Left Pin Names" text box enter a list of pins to appear on the left-hand side of the symbol. Following are some examples of pin lists:

Pin List	Pin Description	Symbol
`A B C`	Three signal pins A, B and C	
`D0..7`	Eight signal pins D0, D1, D2, ... D7	
`DATA[D0..7]`	A bus pin called DATA containing 8 internal pins D0, D1, ... D7	
`CLK,,,D0..7`	Nine pins in all with extra space between CLK and the others	

4) In the "Right Pin Names" text box enter a list of pins to appear on the right-hand side of the symbol, using the same format as above.

5) Click the "Generate" button.

6) Make any desired graphical edits to the auto-generated symbol.

7) If you do not already have a suitable temporary library file open, create one now using the New Lib command (either in the File menu, or using the Parts palette pop-up menu).

8) Save the new part into the temporary library.

See Chapter X - Device Symbol Editing for more symbol editing techniques.

Placing the Block Symbol

Hierarchical block symbols are placed just like any other device symbol:

1) In the Parts palette, use the library selection menu to select the library that you saved the block symbol in.

2) Double-click on the name of the new symbol.

3) Place the symbol where desired on the schematic.

Creating the Internal Circuit

In this section we will take advantage of the fact that DesignWorks creates a default internal circuit template when you open an empty block symbol. It is also possible to create your own internal circuit from scratch. This is discussed below in the section on "bottom-up" block creation.

1) Hold the ⌘ key while clicking on the hierarchical block symbol.

2) Select the Push Into command.

3) A box will appear asking you to confirm that you wish to create a new internal circuit. Click the OK button.

A new circuit window will now open with the default port connectors laid out according to their position on the parent symbol.

Creating a Hierarchical Block - Bottom Up

Any open design can be made into the sub-circuit of a hierarchical block symbol. To create the internal circuit first and work upwards, follow these steps:

1) Create or open the design file that is to become the internal circuit.

2) Create the hierarchical block symbol.

3) Link the sub-circuit to the symbol using the Attach Internal command.

These three steps will be covered in more detail in the following sections.

NOTE: This procedure links the sub-circuit to the symbol only for this design. You can also store the sub-circuit permanently with the symbol in the library. This allows it to be easily used in multiple designs without having to use the Attach Internal command each time. See Chapter X - Device Symbol Editing for details on how this is done.

Creating a Sub-Circuit

Creating a circuit that is to be attached to a parent symbol is exactly the same as creating an independent design, except for the necessity of defining the port interface. Note the following issues when creating the sub-circuit:

• Each signal that is to be an input/output point for the sub-circuit (i.e. a pin on the parent symbol) must have an attached port connector.

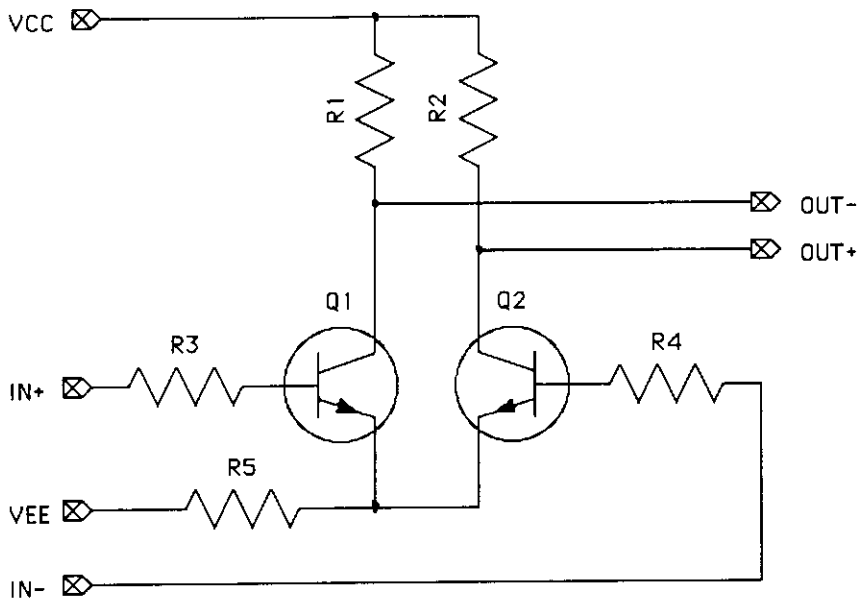
• The name of the port connector device must exactly match the name of the associated pin on the parent symbol.

• For each port, choose a port connector type (input, output or bidirectional) appropriate to the signal flow in your circuit. This allows for more efficient simulation, improved error checking and design clarity.

• Bus ports require special care in creation and naming of port connector symbols. Review the rules given earlier in this chapter carefully.

See the section earlier in this chapter on Port Interface for complete details.

The following sample shows how port connectors are attached to the connection points in a simple circuit:



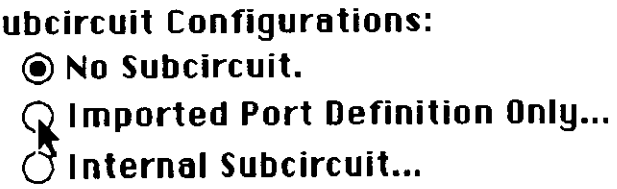
Creating a Symbol for an Existing Sub-Circuit

To create a symbol for a sub-circuit that has already been created an is open in a DesignWorks window:

1) Select the DevEditor tool in the Tools menu.

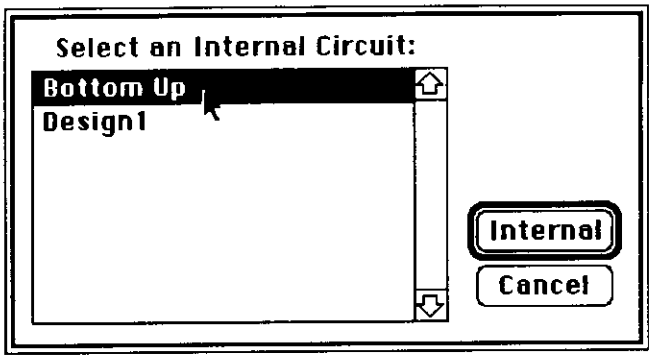
2) Select the Subcircuit & Part Type command in the DevEdit menu.

3) Click on the Imported Port Definition Only button. This will display a list of all currently open designs.



NOTE: Choosing the Internal Subcircuit option instead will cause the circuit definition to be saved in the library with the symbol.

4) Select the desired internal circuit in the list and click the Internal button.



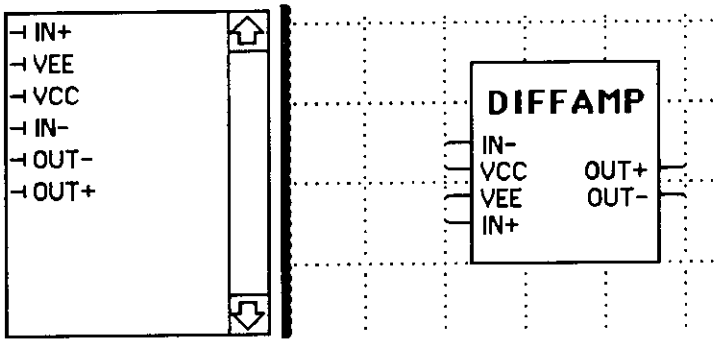
DevEditor now examines the selected circuit and creates a list of pins matching the port connectors in the circuit.

5) Select the Auto Create Symbol command in the DevEdit menu.

6) Click on the Extract Pin List button.

This will place input and output pins in default positions in the Left and Right Pin Names boxes. You may move names between these boxes as desired to determine placement of pins. Be careful not to add or delete any names. You can also change the part name that will appear in the symbol, if desired.

7) Click on the Generate button. A default symbol will now appear in the DevEditor window.



This symbol can be edited manually using the drawing tools at the left.

8) Create a temporary library to save the symbol in. This is done using the New Lib command in the Libraries sub-menu of the File menu, or in the Parts palette pop-up menu.

9) Select the Save Part As command in the File menu. Select a suitable part name and double-click on the new library that was created in the last step.

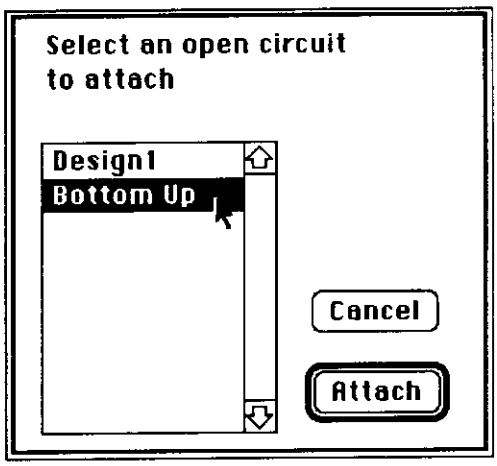
10) Close the DevEditor window.

Linking a Symbol to an Existing Sub-Circuit

1) Place the new symbol in the master circuit for the design, by double-clicking on it in the Parts palette and placing it in a suitable position in the schematic.

2) Click on the new symbol to select it.

3) Choose the Attach Internal command in the Options menu. A box will appear listing all open designs except the current one (a sub-circuit cannot be attached inside itself!):



Select the desired internal circuit (presumably the same one that was selected while in the DevEditor), and click on the Attach button.

The internal circuit will now be linked to the selected symbol and its window brought to the front.

NOTE: The Attach Internal command in effect changes the definition for the selected device by adding an internal circuit. If there were other devices of the same type on the schematic, you will be asked if you want to update them when the internal circuit is closed.

Changing the Hierarchical Name Separator

The separator character used to generate hierarchical names is stored in the design attribute field "HierNameSep".

To change it:

1) Select the Set Design Attributes command in the Options menu.
2) Select the HierNameSep field in the field list.
3) Enter a new value in the data box.
4) Click OK.

Chapter VIII - Before Starting a Major Design

This chapter provides information that should be considered before starting any major design. Correct choices made in advance concerning usage of hierarchy mode, attribute format, sheet layout etc. will save time and frustration later. Issues such as hardcopy format, drawing standards, interfacing with other systems, and reports required are addressed.

This chapter is not intended as a specific reference for any DesignWorks features. It focuses on your desired end-product and how best to make use of this package to achieve that goal. Its primary aim is to raise issues that you should consider and alternatives that you may wish to test before proceeding.

In addition, this chapter makes only general reference to third-party products and specific issues on interfacing with them. Capilano Computing has a number of Technical Notes available on issues concerning specific third-party packages. Some of these notes were provided with this manual and others can be obtained by calling our Technical Support department at the number listed in the front of this manual.

Hierarchical Design

Hierarchy Mode

DesignWorks offers three hierarchy modes to suit a variety of applications. Following are a number of points to consider before settling on which to use.

Hierarchical design is a powerful technique, but does add some complexity to the design process. If you are one of the following:

- a new user of schematic capture tools
- working with small-to-medium designs (say, less than 200 ICs)
- interfacing to a third-party Printed Circuit Board layout package

...then we recommend sticking with "Flat" mode and not using hierarchical blocks in your designs. Once you have gained some experience with creating schematics in DesignWorks and moving design data between systems, you can review the hierarchy issues discussed in this manual.

Pure Hierarchy Mode Restrictions

Pure hierarchy mode is intended for large designs that:

- lend themselves to a very structured hierarchical definition.
- do not require device instance data.
- do not require flat netlist output.

Unless you are sure your design meets these requirements, do not use Pure mode. Pure mode is not recommended for printed circuit board-level designs.

Hardcopy Output Format

The size and format of the schematic sheets you use will likely be determined by your organization's drawing standards and by what form of hardcopy device you have available. It is best to do some test sheets in your chosen format before proceeding with a large design. Reformatting a design to a different sheet size is an unpleasant task.

Printing

Most laser or dot-matrix printers have a number of options for print area size and scaling. For this reason, it is best to either:

- use the "Use Page Setup" option and default borders in the Custom Sheet Info command. This way, the schematic drawing area will automatically adjust to changes in printer page setup.

OR

- choose a format and scale factor that looks good on your printer and create a fixed sheet template for it.

Plotting

Following are a number of issues to consider before creating a design that will output to a plotter.

- If you create any device symbols using an external drawing package, be sure not to use any "paint" or "bitmap" features. Any bitmapped graphic objects will plot poorly, if at all, on a pen plotter.
- If you are going to be plotting many multipage drawings, you may wish to purchase a third-party pen plotter package that installs as a printer driver. This will allow you to plot your entire design with a single command.

Interfacing with PCB Layout Systems

If you plan to use DesignWorks in conjunction with any other package for Printed Circuit Board layout, it is best to review any restrictions that may be imposed on data imported to that system. A number of possible considerations are mentioned below.

> IMPORTANT: We strongly recommend that you create a small test case and go through the procedure of creating the output files and passing them to the desired target system. This will help you correct interface problems before they have a major impact on your design.

Naming restrictions

DesignWorks allows devices and signals to have any name up to 16 characters. Except for carriage returns, anything you can type on the keyboard can be part of a name. Many systems place some restrictions on names, for example:

- length restrictions
- usage of special characters (very few systems will accept Option-key characters used on the Macintosh. These are not standard ASCII codes).
- case sensitivity

If you intend to use DesignWorks' hierarchical design features, you will need to determine whether hierarchical names (e.g. "BLOCK1/INBUF/REG1") of arbitrary length are acceptable. If not, it will be necessary to ensure that each device and signal has a unique identifier.

Part Names and Package Codes

Most netlist formats for PCB systems contain either or both of the following for each device in the design:

- a part name, e.g. "74ALS244"
- a package code, e.g. "DIP20"

There is no clear industry standard for the format of either of these codes. It is therefore important to verify that the type names and package codes used in the DesignWorks libraries will be acceptable to the target system. If not, it will be necessary to either update the library entries for the parts you will be using or make appropriate attribute entries for each device as you use it on the diagram.

Interfacing with SPICE

DesignWorks 3.0 has a number of features specifically intended to assist in creating designs for SPICE-based analog simulators.

See also the SPICE report examples in Chapter XII - Report Generation.

SPICE Data Field

An attribute field called "Spice" is predefined for devices and the design itself. If the standard Report forms are used, any text data entered in this field is included in the output file, as follows:

• for devices, the data is appended to the corresponding device line in the file. E.g. entering the value "SIN(0.0 0.01V)" in the Spice field for a voltage source would create the following line in the SPICE output file:

```
V1 0 VPLUS SIN(0.0 0.01V)
```

• for the design, the contents of the field is inserted verbatim at the front of the file. This can be used to enter parameter lines, model definitions, etc. Multiple lines can be entered if desired.

SPICE Name Prefix

Library parts in the SPICE Devices standard library have a field called "Name.Spice" defined. This field contains the standard SPICE name prefix letter for that type. To use this prefix, use the Design Attributes command to set the PrefixField field to "Name.Spice".

Hierarchical Designs for SPICE

Hierarchical designs created in Pure or Physical hierarchy mode will be written to the SPICE file using the SUBCKT statement. See the SPICE sample designs provided with the package.

Pin Ordering for SPICE

In the SPICE netlist format, the functions of device pins are determined by the order of appearance in the file. Therefore, the order of placement of pins on a device symbol is extremely important for this usage. Pins are normally placed in the order they appear when the symbol is opened using

the DevEditor, although this can be modified using the Report tool's DEVPINSEQUENCE facility and the PinSequence attribute field, described below.

Discrete Component Symbols

Discrete devices in the standard SPICE Devices library provided with DesignWorks have their pin order preset for SPICE usage. If you are using any other discrete component symbols, or creating your own, check the pin order carefully.

Digital Component Symbols

Unfortunately, there is no established standard pin ordering for digital simulation extensions to SPICE. The standard DesignWorks digital libraries are not set up to work with SPICE-based digital simulators, although most gate-level parts are created using the natural order of inputs first, then outputs.

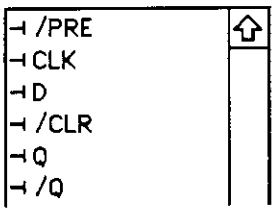
In order to use digital components with a SPICE format, it will be necessary to either:

- enter a list of the pins to be listed in the file in the device's PinSequence attribute field, OR
- create a special symbol with the pins in the correct order, OR
- obtain libraries of symbols specially created for use with your simulator.

Using the PinSequence Field

The standard SPICE report forms provided with DesignWorks make use of the PinSequence attribute field to specify pin ordering for digital parts. To specify pin ordering, it is first necessary to know the names of all the pins on the device in question. This can be obtained using the Get Info command, the Pin Info command or by opening the part type using the DevEditor.

For example, the 74ALS74 part has the following pin list when viewed in the DevEditor:



To specify an alternate order for output, you could enter the following string into the PinSequence field for the device:

```
/CLR,D,CLK,/PRE,Q,/Q
```

Only the items entered in this list will appear, i.e. not all pins need be listed and some can be listed more than once, if desired. The pin names must match exactly the list as seen in the DevEditor, including case. Unrecognized names are ignored by the Report tool.

NOTE: Entries in the PinSequence field are effective only if specifically enabled by the selected Report form file. The SPICE form files provided with DesignWorks include this entry. If you are creating your own format, refer to the description of the $DEVPINSEQUENCE format command.

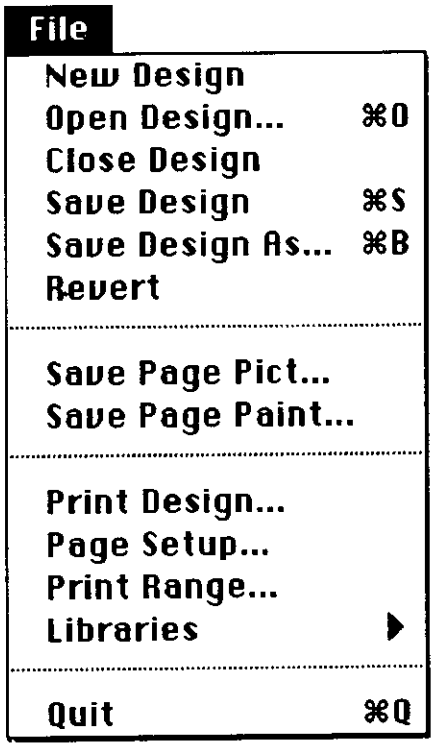
Chapter IX - Schematic Menu Reference

This chapter provides detailed information on the use of individual menu commands in DesignWorks. In order to give you rapid access to commands, the program has two different sets of menus:

- Standard Menus - these are the normal File, Edit and other menus that appear in the menu bar at the top of the screen.
- Pop-up Menus - these menus will appear anywhere on the schematic diagram when the mouse is clicked while holding the ⌘ key pressed. Most of the commands appearing in these menus also appear in the standard menus, so they can be considered a short-cut. The type of menu that appears depends on where the mouse is clicked. Different menus appear for Circuit, Signal, Device, Pin and Attribute functions.

The commands in the DevEdit menu are covered in Chapter X - Device Symbol Editing.

File Menu



NOTE: Some of the items in the File menu may change text when other tools are in use (e.g. the DevEditor). The functions described here apply only to the Schematic tool.

New Design

This command will create a new, empty design window. This can be used to create an entirely new design, to create a sub-circuit that will later be associated with a device, or as a temporary area to edit a circuit scrap.

There is no fixed limit on the number of designs that can be open at once, although the complete contents of all open designs must fit into memory.

The design will use the current printer Page Setup for its sheet size. More information on these settings is provided under the Custom Sheet Info and

Design Preferences commands. New Design does not create a disk file and has no effect on any files on your disk.

Open Design...

This command allows a design to be opened from a disk file.

When you open a file, the circuit data is read into memory in its entirety and no more access to the disk file is required. DesignWorks will let you open multiple copies of the same file and makes no attempt to restrict you from writing any of them back to the same file. If you do this, it is up to you to keep track of which windows have been updated and what file you want to save them into.

Compatibility With Older Versions

DesignWorks 3.0 will not directly read files created by older versions. They must be first converted using the Converter tool described in another manual.

Close Design

Close Design closes all the circuit windows associated with the current design and removes all data from memory. If any changes have been made to your design since the last Open or Save, then you will be asked if you wish to save those changes. The same effect is achieved by clicking the "Go Away" box in the upper left corner of the last circuit window.

Save Design and Save Design As...

Save Design and **Save Design As** save the current circuit in a circuit file. **Save Design** saves the circuit back into the file that was most-recently opened. It will be disabled if no file has been opened. If you select **Save Design As**, a dialog box will be displayed requesting the name the new file. The default name will be the current title of the circuit window, i.e. the name of the most recently opened or saved file.

Revert

This command rereads the current design from the disk file it was last saved to or read from. If any changes have been made since the last Save, you will be prompted to confirm the choice before they are discarded.

Save Page Pict...

DesignWorks can save the circuit diagrams in the Macintosh standard PICT graphics format. This capability allows you to pass graphics to other programs for plotting, enhancement, or incorporation into other documentation.

PICT format is a standard format for representing graphics objects such as lines, circles, text, etc. This format can be read by most Macintosh-based graphics and drafting packages and pen-plotter drivers. PICT file format does not restrict the size of the diagram. Note that DesignWorks cannot read these files, it can only create them.

Saving the Diagram as a PICT File

Save Page Pict stores the circuit page in the topmost circuit window to a PICT format file. Note that variations in coordinate systems for different programs may cause the diagram to be shifted off to one side when it is read by another graphics program. This can usually be remedied by selecting everything from within the draw program and centering in the drawing space. The border and background grid will appear or not appear in the file according the "Print Drawing Grid" and "Print Default Border" options in the Custom Sheet Info command.

Plotting the Circuit Diagram

The circuit diagrams can be plotted from the PICT file using a separate plotter driver package. If you purchased the plotter driver with DesignWorks, then refer to the manual for that program in the Utilities section of the binder. Otherwise you can use plotter drivers available from other sources.

See Chapter V - Advanced Schematic Editing for more information about setting page sizes and scaling for plotter output.

Print Design...

Print Design allows you to print all or part of your circuit diagram. If the diagram will not fit on a single page, it will be broken into as many parts as are needed, based upon paper size specified in Page Setup. You can preview the page breaks by using the Show Page Breaks option in the Design Preferences command. For purposes of specifying a range to print, pages are numbered from top to bottom, then left to right, starting on circuit page 1. In multi-page circuits the print pages are numbered sequentially from the top left corner of circuit page 1 to the bottom right corner of the last circuit page. Page numbers do not appear in the printed output.

In hierarchical designs, pages are numbered separately for each circuit level. The Print Range command is used to determine printing extent.

ImageWriter Notes

DesignWorks diagrams can be printed successfully on an Apple ImageWriter printer using any of the available scaling and orientation options. Draft mode will cause only the text in the diagram to be printed, which is normally of limited value.

LaserWriter Notes

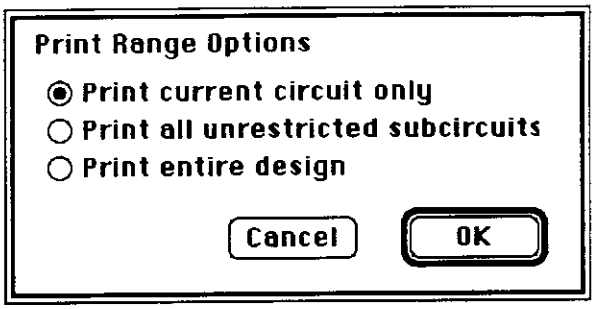
DesignWorks diagrams can be printed successfully on a LaserWriter printer (or other laser printer), but a number of precautions must be taken into account. In particular problems can arise with user-created device symbols, depending on how the symbols were drawn. Bit-mapped graphics, such as are produced by "paint" type graphics programs, may not scale smoothly and should be avoided.

Page Setup

The page setup dialog will be presented allowing you to choose the size and orientation of printer paper you wish to use. Once chosen, this information will be stored with your design file and affect the page outlines shown in the command and the Show Printed Page Breaks option in the Design Preferences command.

Print Range...

This command is used to set the scope of printing in a hierarchical design.



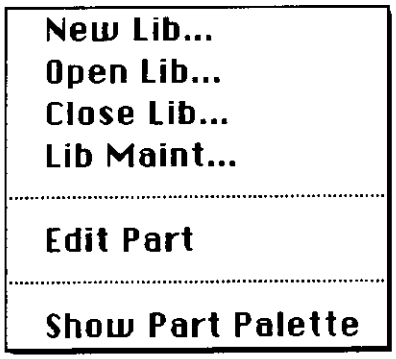
Three options are available:

1) Print current circuit only: With this setting, the Print Design command will print only pages belonging to the circuit level in the current window. With this setting, the page numbers shown by the "Show Printed Page Breaks" option can be used to select a subset of pages to print.

2) Print all unrestricted circuits: With this setting, the design's master circuit and all unlocked internal circuits will be printed. (The "locked" setting is controlled by a check box in the Get Info command for the parent device.) With this setting, the page numbers shown by the "Show Printed Page Breaks" option will not be relevant since they are numbered separately for each internal circuit.

3) Print entire design: Print the design's master circuit and all internal circuits, regardless of "locked" status.

Libraries Sub-Menu



This menu can be reached in either of the following ways:

1) Slide down the File menu to the Libraries sub-menu. The sub-menu appears to the side of the main menu.

2) Hold the ⌘ key pressed on the keyboard and click anywhere in the Parts palette.

New Lib...

The New Lib command allows you to create a new, empty symbol library file on your disk. It will be opened automatically and appear in the Parts palette.

Open Lib...

The Open Lib command allows you to select an existing symbol library file to be opened. The name of the library will appear in the pop-up library selection menu in the Parts palette. A small amount of memory is occupied by each open library file.

Note: Libraries can be opened automatically when the program starts by placing them in the default library folder or by using the LIBRARY and LIBRARYFOLDER setup file keywords.

See Chapter II - Getting Started for more information.

Close Lib...

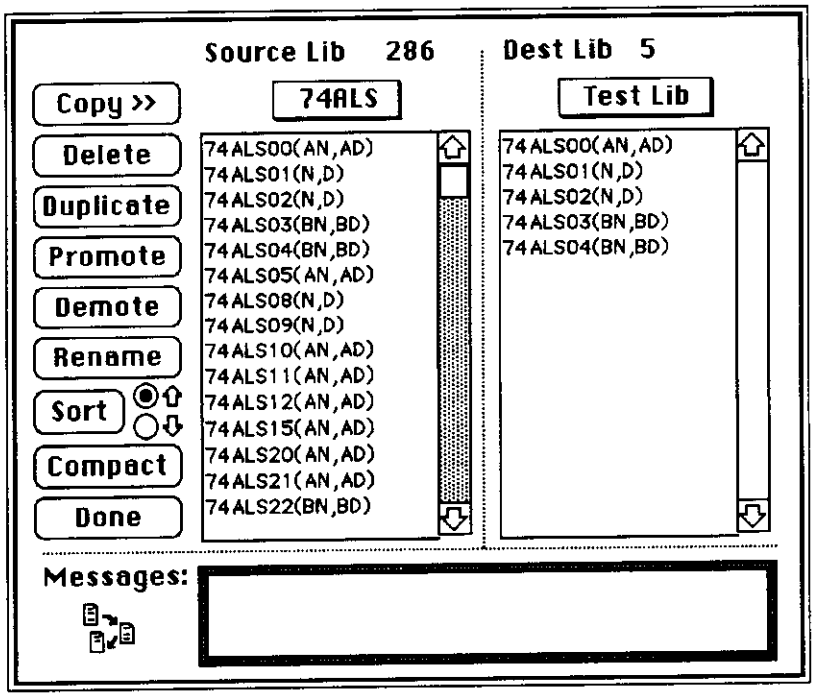
Close Lib allows you to close an open library and remove it from the Parts palette. Any information required for parts used in any open designs will be automatically retained in memory. When you select the Close Lib command, a box will appear listing the open libraries. Simply pick one by clicking on it and then press the Close Lib button, or simply double-click on the name of the library.

Edit Part

This command starts up the DevEditor tool and opens the part currently selected in the Parts palette. This item will be disabled if nothing is selected in the Parts palette.

Lib Maint...

This command invokes a variety of library maintenance functions. The following box will be displayed:



The following table summarizes the options available.

Source Lib	The Source Lib is the primary library operated on by all the command buttons. The Source Lib can be any one of the currently open libraries, selected using the pop-up menu at the head of the list. Any single item in the Source Lib can be selected by clicking on it. In contiguous set of items can be selected by holding the Shift key to add to the selection.
Dest Lib	The Dest Lib is used only as the destination of the Copy command. No items can be selected in this list.
Copy	This button causes the selected parts in the Source Lib list to be copied to the Dest Lib.
Delete	This button causes the selected parts in the Source Lib to be deleted. THIS CANNOT BE UNDONE!!!
Duplicate	This causes the selected parts in the Source Lib to be duplicated, i.e. a copy of the selected items is made in the Source Lib. The Dest Lib is not affected.
Promote/Demote	The Promote and Demote buttons cause the selected items in the Source Lib to be moved up or down the list, respectively.
Rename	Rename displays a box allowing a new name to be entered for the selected part.
Sort	This button sorts the entire list alphabetically. The adjacent arrow buttons determine the direction of the sort.
Compact	This button causes the Source Lib to be compacted to the destination lib (which must be empty), i.e. any free space due to deletions is removed. See more information below.
Done	The Done button closes the Lib Maint dialog box.

Library Compaction

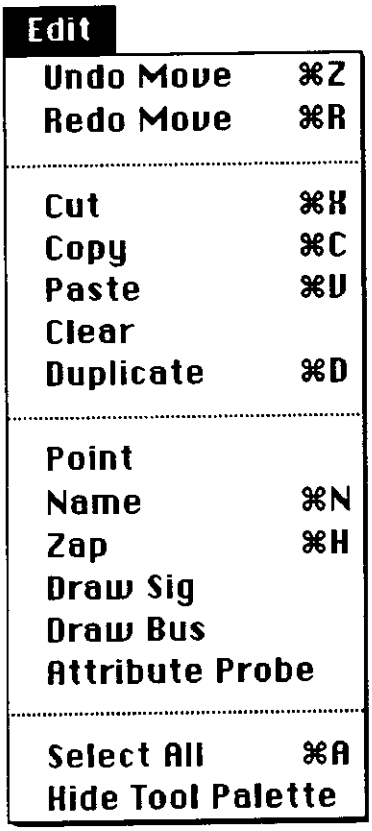
When parts are deleted from a library the free space in the file is not automatically recovered. In most cases this is not a significant overhead.

However, if a large percentage of the parts in a library have been deleted then you may wish to compact the file. To do this:

1) Create a new, empty library which will become the target for the Compact operation.
2) Select the Lib Maint command.
3) Select the library to be compacted as the Source Lib.
4) Select the new, empty library as the Dest Lib.
5) Click on the Compact button.

IMPORTANT: Verify that the new destination library is correct before discarding the old copy.

Edit Menu



Undo

This command undoes the last editing operation that was performed. The text of this menu item will change based on the type of operation. Generally, only schematic editing operations can be undone. Major structural changes, such as Define Attribute Fields, or any menu commands involving a dialog box are usually not undoable.

Undo never changes the contents of the Clipboard. E.g. after a Cut operation, Undo will restore the schematic, but leave the Cut objects on the Clipboard.

Redo

This command redoes the last Undo command. It will only be enabled immediately after an Undo operation. Any other editing operation will disable this item.

Using the Clipboard

The standard Macintosh clipboard commands Cut, Copy and Paste can be used to move or copy circuit fragments, graphical and text information within a single circuit window, between multiple windows, and between different programs (e.g. word processing or drafting).

Using Clipboard Data From Other Programs

When you enter DesignWorks, the clipboard may contain graphical or text information Cut or Copied from a document in another program. DesignWorks allows you to make use of this information in two ways:

- a Macintosh picture can be Pasted directly onto your diagram in any desired position. Once placed, it can be dragged, Zapped, Cut, Copied, Duplicated, etc. using the various editing commands available for other circuit elements. When a picture item is selected, the Get Info command can be used to select an optional border. A graphics object is considered to be a monolithic item on a schematic and can only be moved or deleted in its entirety. It can, however, be copied and pasted to the DevEditor tool to take advantage of the standard drawing tools available there.
- text information from a word processor or text editor can be pasted into a text block. See more information under the Text command below.

See more information in Chapter IV - Schematic Editing.

Using Clipboard Data From DesignWorks

When a Cut or Copy is done, two types of data are placed on the clipboard:

• A Macintosh picture ("PICT" format) of the selected items, which can be pasted into a graphics document using most drawing programs.

• The DesignWorks circuit info for the selected items. This data is in a format that only DesignWorks can understand and is discarded when you Quit.

WARNING: Circuit strcutural information on the clipboard is discarded when you quit the program. Only picture and text data is retained. You cannot Copy and Paste circuit data between DesignWorks sessions.

Cut and Copy work on the currently selected group of circuit objects and will be disabled if no objects are selected. When items are copied onto the clipboard, their names are copied with them, which may result in duplicate names. If duplicate signal names are pasted back into the circuit page they were copied from, then logical connections will be made between the like-named segments.

Cut

Cut removes the currently selected objects from the circuit and transfers them to the clipboard. It is equivalent to doing a Copy and then a Clear. Cut will be disabled if no objects are selected.

Copy

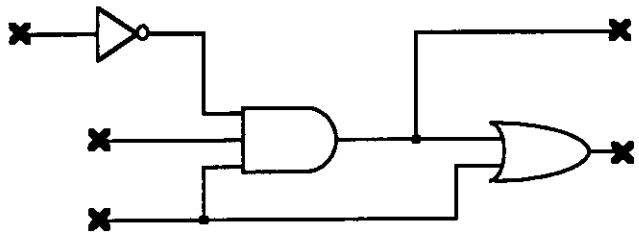
Copies the currently selected objects onto the clipboard without removing them. This can be used to duplicate a circuit group, copy it from one file to another, or to copy a picture of the circuit group to a drawing program. See the notes on clipboard data above. Copy will be disabled if no objects are currently selected.

Paste

Paste replaces the cursor with a flickering image of the contents of the Clipboard in the circuit window. As noted above, this data may be a circuit group copied from within DesignWorks, or it may be text or picture information created by another program or module. The image of the clipboard data can be dragged around and positioned as desired. The item will be made a permanent part of your diagram when the mouse button is pressed.

DesignWorks checks for signal connections only at "loose ends" in the signal lines being pasted, i.e. ends of line segments which do not touch devices or other line segments. E.g. if the following circuit scrap was

pasted, the points marked X would be checked for connection to the existing circuit.



NOTE: Connection "hit testing" can be disabled by holding the Option key on the keyboard depressed while clicking the mouse button (this also applies to single device placing). In this case, the circuit scrap is placed, but no connections will be made to adjacent items. This allows the group to be selected again (by Option-double-clicking on any device in the group) and moved without interactions with other objects in the circuit.

Paste will be disabled if there is no information on the clipboard of a recognized type.

Rotation on Paste and Duplicate

Any group of objects being Pasted or Duplicated can be rotated using the same controls as when placing a device:

- the Orientation pop-up menu in the tool palette.
- the arrow keys on the keyboard.

Note that these controls are only effective while actually moving the flickering image of the object being pasted. Each Paste or Duplicate always starts in the same orientation as the source.

NOTE: The Orientation command cannot be used for Paste or Duplicate since selecting this menu command will abort the paste operation.

Clear

Removes the currently selected objects from the circuit. Typing the backspace or delete key on the keyboard is equivalent to Clear. This will be disabled if no devices or signals are selected.

Duplicate

Makes a copy of the selected circuit group which can be dragged and positioned as desired. This is equivalent to selecting Copy and then Paste, except that the selected circuit scrap is not placed on the clipboard for future use. See the notes under Paste, above, on how connections are made when a group is placed in the circuit. Note that the duplicated objects can be rotated using the orientation pop-up menu in the tool palette or the arrow keys on the keyboard.

Point

This selects the normal operating mode for DesignWorks, indicated by the arrow cursor. Selecting this command is equivalent to clicking on the icon in the tool palette. The following functions are accessible in Point mode:

- by clicking on an object you can select the object for operations using the Edit menu commands. To select an I/O device, or to select multiple objects, hold the SHIFT key on the keyboard while you click. To select a Background object, hold the Command and Option keys while clicking on it.
- by clicking and dragging near the end of a device pin or signal line you can extend that line in any direction.
- by clicking and dragging a signal line anywhere but near the end then you can change its perpendicular position.
- by clicking and dragging any other object you can reposition the object.

All of the above functions are described in more detail in Chapter IV - Schematic Editing.

Shortcuts to Point

Since it is frequently desired to return to Point mode, two short cuts are provided:

- clicking the mouse button while pointing anywhere inside the menu bar
- pressing the spacebar on the keyboard.

Text

The Text command changes the current cursor to Text mode and is equivalent to clicking on the A icon on the tool palette. In this mode the following functions are available:

- a name can be associated with a device by clicking and holding on the device in question, then moving the cursor to the desired position for the text then releasing the button, then typing the desired name (up to 16 characters) followed by Enter.
- a name can be associated with a signal by clicking and holding anywhere along a signal line, then proceeding as for devices above. Signal names differ from device names in that they can appear at multiple locations along the length of the signal line, up to a maximum of 100 positions. Additional name positions are added by simply repeating the naming procedure as many times as required. If the name at any position is altered then all positions are updated.
- any existing attribute item displayed on the schematic (including a name) can be editing by clicking in it. If the text in question was being displayed with the field name or if it was rotated, then an edit box will be displayed, otherwise it can be edited right on the diagram.
- a pin number can be placed on a device pin by clicking on the pin within 5 screen dots of the device. A blinking insertion point will appear, and you will be able to type up to 4 characters. Press the Enter key to terminate the pin number.
- random text (i.e. not associated with a specific device or signal) can be placed anywhere on the diagram by clicking on the diagram other than on a device or signal. This text can contain carriage returns or any other characters.
- any of the above items can be edited by clicking anywhere in the existing object. The blinking insertion point will appear in the text at the position of the click.

All of the above functions are described in more detail in Chapter IV - Schematic Editing.

Zap

The Zap command changes the current cursor to Zap mode and is equivalent to clicking on the icon in the tool palette. When the tip of this cursor is

clicked on any object in a circuit then that object is removed. The Zap cursor is more selective than using the Clear command on selected objects:

- Signal or bus lines: only the line segment under the cursor is removed.
- Pin numbers: Removes the pin number.
- Attribute items: Removes the visible attribute text from the diagram, but leaves the value associated with the object.

See more information on this command and other editing features in Chapter IV - Schematic Editing.

Draw Sig

The Draw Sig command is equivalent to clicking on the + icon in the tool palette and places the program in signal drawing mode. In this mode you can draw or extend signal lines as follows:

- clicking anywhere along an existing signal line extends the signal, starting at that point.
- a new signal can be created by clicking anywhere on the diagram.

When you click again, the lines on the screen become permanent and a new set of lines are drawn starting at that point. A number of line routing options are selected by pressing the Command, Option and Shift keys while drawing. To terminate signal drawing mode, double-click, press the space bar or click anywhere in the menu bar.

See more information on signal drawing modes in Chapter IV - Schematic Editing.

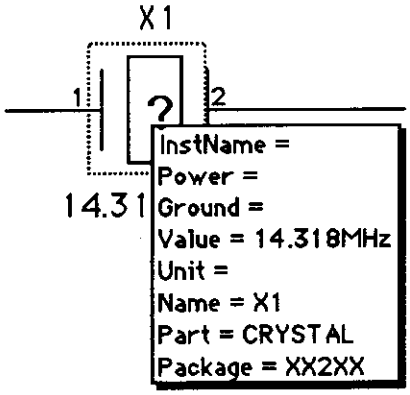
Draw Bus

The Draw Bus command is equivalent to clicking on the + icon in the tool palette. Bus drawing mode behaves exactly like signal drawing mode except that a bus line is created instead of a signal line.

Attribute Probe

The Attribute Probe command displays the ? cursor and is equivalent to clicking on the ? tool in the tool palette. In this mode, clicking on an

object displays a small window containing the contents of attribute fields associated with the object, as in the following example:



NOTE: The dot at the base of the ? icon is the "hot point" that determines which object is selected.

For compactness, only "primary" fields are shown and only the first 32 characters of the field value will appear. Whether or not a field is considered "primary" is determined by the setting of the "Include in Primary List" switch in the Define Attribute Fields command.

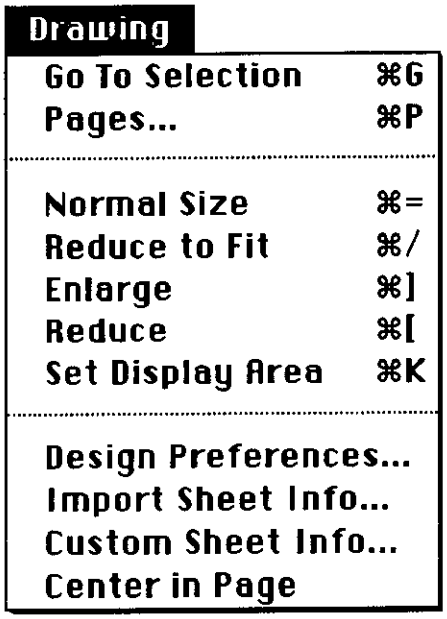
Select All

This command selects and highlights all elements in the current circuit page, except picture and text items marked as Background items. (Background objects can be selected by holding the Command and Option keys pressed while clicking on them.) The entire page can then be operated on with the clipboard commands and others that work on selected items.

Show Tool Palette/Hide Tool Palette

This command hides or displays the floating tool palette window. It can also be hidden by clicking in its "go-away" box at the top left corner.

Drawing Menu



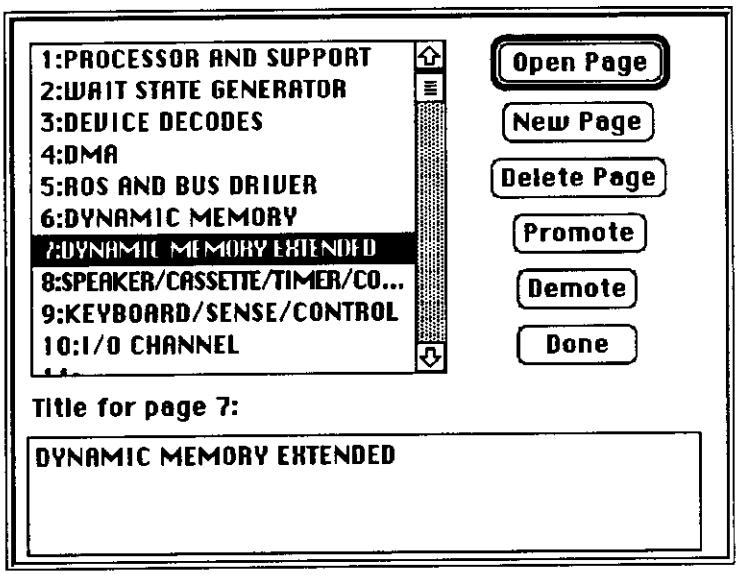
The Drawing menu contains commands for positioning and scaling the drawing on the screen, and for setting sheet size, display and printing options.

Go To Selection

This command causes the circuit position and scaling to be adjusted so that the currently selected items are centered and just fit in the circuit window. The scaling will be set to 100% maximum.

Pages...

The Page command is used to create a new page, display an existing page, change page order, or set a page title. Executing this command will produce the following box:



Operations on pages are performed by selecting the pages in the scrolling list and clicking on one of the operation buttons. More than one page can be selected at a time (for all operations except setting the page title) by holding the command or shift keys pressed while clicking in the list.

WARNING: Operations in this box cannot be undone!!!

Open Page

The Open Page button causes the pages selected in the list to be displayed in a window. If a window is already open with the selected page, then it is brought to the front.

New Page

The New Page button causes a new, empty page to be added to the circuit and opens it in a new window.

Delete Page

The Delete Page button causes the page selected in the list to be deleted from the circuit. Note that the page must already be empty, i.e. all objects must have been removed from the page (for example using the Select All and Clear) commands before this will be allowed. Higher numbered pages will be renumbered when this page is removed.

Promote/Demote

The Promote and Demote buttons allow you to change the number order of the pages. Promote causes all pages selected in the page list to be moved one step toward the front of the list, i.e. their page number will be reduced by one. All non-selected pages will be renumbered as needed. Demote similarly moves all selected pages toward the end of the list. Promote will be disabled if the first page is selected and Demote will be disabled if the last page is selected.

Page Title

The page title text box allows a title of up to 63 characters to be entered for the selected page. This title can be accessed via the $PAGETITLE variable in text boxes.

See Chapter IV - Schematic Editing for more information.

Screen Scaling Commands

Four commands are provided which control the enlargement or reduction of the circuit diagram on the screen. These commands control screen display only and have no effect on the stored circuit information, printed output, or graphics files. The maximum enlargement allowed is 200% and the minimum level of reduction is 20% of normal size. Due to the integer calculations that are done by DesignWorks and by the Macintosh system, device symbols and text may be displayed rather crudely at scale factors other than 100%. It is best to do most editing at normal size to ensure that everything lines up as you would expect.

Normal Size

Normal Size sets the screen scale to 100%.

Reduce to Fit

Reduce to Fit sets the scale factor and centers the display so that the entire diagram fits on the screen. If the size of the diagram and the size of the window is such that this would require a scale factor of less than 20%, then the scale is set to 20% and the diagram is centered. If the diagram fits completely in the window at 100%, then the scale is set to 100% and the diagram is centered.

Enlarge

Enlarge increases the scale factor by about 20%, causing the diagram to appear larger on the screen. If this causes the setting to exceed the maximum of 200%, it is set to the maximum.

Reduce

Reduce decreases the scale factor by about 20%, causing the diagram to appear smaller on the screen. If this causes the setting to go below the minimum of 20%, it is set to 20%.

Magnify

This command provides an alternative method of zooming into and out of a selected area of the diagram. When you select Magnify, the cursor changes into the shape.

Zooming In

Two methods of zooming in are provided:

1) Clicking and releasing the mouse button on a point on the diagram will zoom in to that point by one step.

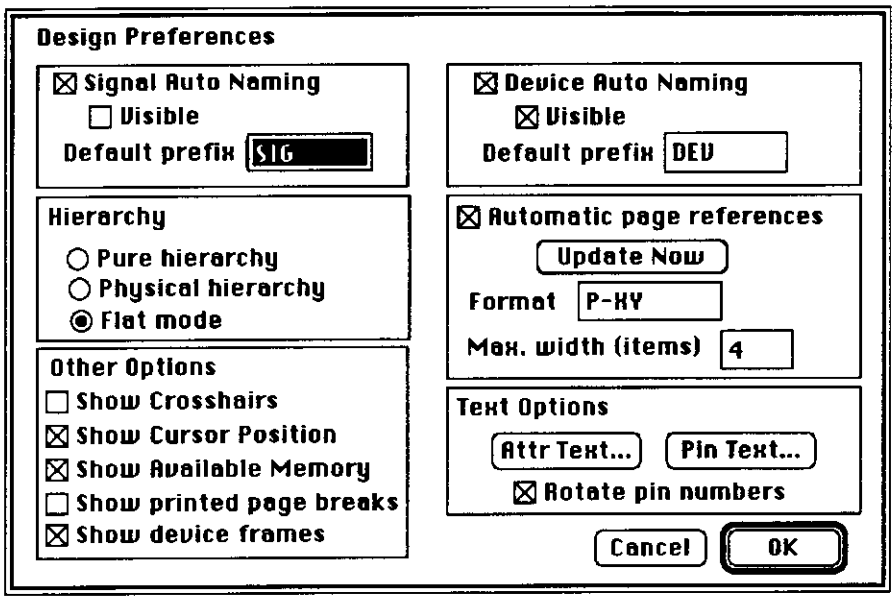
2) Clicking and dragging the mouse down and to the right zooms in on the selected area. The point at which you press the mouse button will become the top left corner of the new viewing area. The point at which you release the button will become approximately the lower right corner of the displayed area. The circuit position and scaling will be adjusted to display the indicated area.

Zooming Out

Clicking and dragging the mouse up and to the left zooms out to view more of the schematic in the window. The degree of change in the scale factor is determined by how far the mouse is moved. Moving a small distance zooms out by one step (equivalent to using the Reduce command). Moving most of the way across the window is equivalent to doing a Reduce to Fit.

Design Preferences...

The Design Preferences command is used to set a number of options which have global effect throughout a design. Selecting this command displays the following box:



Signal Auto-Naming

When this option is enabled, all signals and busses will be given a default name when they are created. This ensures that every signal in a design has a unique name and can be identified in a netlist.

If the "Visible" option is selected, the default name will be displayed on the diagram adjacent to the signal when it is created. This is normally not recommended as it creates clutter on the diagram while creating and editing signals.

See more information on this topic in Chapter IV - Schematic Editing.

Device Auto-Naming

When this option is enabled, all devices (except pseudo-devices) will be given a default name when they are created. This ensures that every device in a design has a unique name and can be identified in a netlist.

IMPORTANT: If the design is in "flat" hierarchy mode (the default) and auto-packaging is enabled (the default), then Device Auto-Naming will have no effect. In "flat" mode, both the Packager and the Device Auto-Naming feature set the Name attribute field. In this case, the Packager takes precedence.

If the "Visible" option is selected, the default name will be displayed on the diagram adjacent to the device when it is created.

See more information on this topic in Chapter V - Advanced Schematic Editing.

Hierarchy Options

The Pure Hierarchy, Physical Hierarchy, and Flat Mode buttons allow you to select the hierarchy mode for the current design.

WARNING: Changing the hierarchy mode in a non-empty design can cause some data stored in the design to be permanently deleted. Make sure you understand the impact of your choice before proceeding.

If you make a hierarchy mode change that will cause data to be discarded, you will be warned and allowed to cancel the operation.

See Chapter VII - Hierarchical Design for more information.

Page References

Automatic page reference updating is enabled whenever the Automatic Page References box is checked. This means that page reference text will automatically be placed next to any named Page Connector that is placed in the design, and will be updated whenever any page connector is moved.

If the Automatic Page References box is unchecked, page references will be updated only when the Update Now button is clicked. This can be used for large designs where page reference updating may cause delays while editing.

Page Reference Format

Two aspects of the page reference format can be controlled:

- the "Format" text item controls the appearance of each page reference in the reference list.

• the "Max. Width" item controls the number of references that will appear on each line before creating a new line. For signals with many connections, this prevents unwieldy page reference strings.

Three characters are special in the format string:

P	will be replaced by the page number
X	will be replaced by the X grid position
Y	will be replaced by the Y grid position

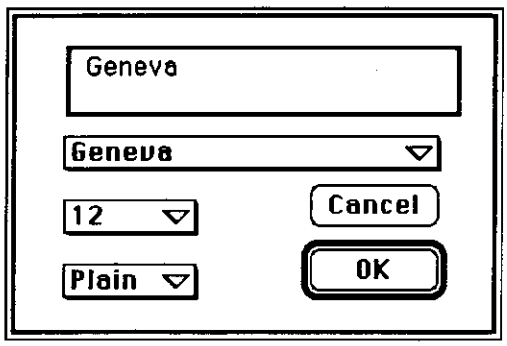
The default format is "P-XY" which generates pages references like "2-C4" (i.e. page 2, grid position C4). All characters other than the special ones above will be placed in the page reference list verbatim.

The maximum length of a format string is 16 characters.

See more information in Chapter V - Advanced Schematic Editing.

Attribute Text Options

Clicking on the "Attr Text..." button will display the following text specification dialog box:



This box allows you to select the text font, size and style used for all attributes displayed in the design, including:

- Device and signal names.
- Bus breakout and bus pin labels.
- All other attributes displayed on the diagram.

but not including:

- Border text (set using the Custom Sheet Info command).
- Pin numbers (set using the Pin Text... button).

The text changes are applied when the OK button in the Design Preferences box is clicked. For larger designs, there may be a substantial delay while new positions of all displayed attribute items are calculated.

Pin Text Options

The Pin Text... button displays the same text specification box shown above for Attr Text. Any changes made to this text style will be applied to all pin numbers displayed throughout the design. The text changes are applied when the OK button in the Design Preferences box is clicked.

Allow Rotated Pin Numbers

If this item is checked, pin numbers displayed on all north- and south-facing pins will be rotated 90° to run along the length of the pin. If this item is unchecked, all pin numbers will be displayed horizontally adjacent to the pin.

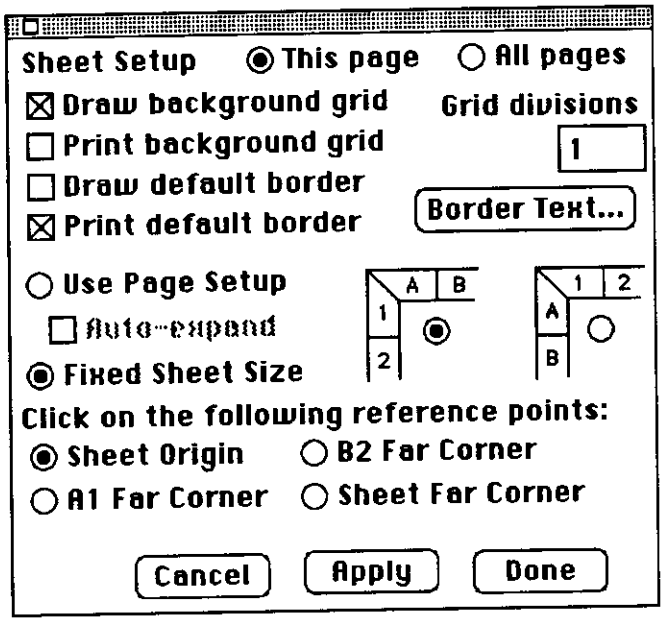
Import Sheet Info...

This command allows you to import a standard sheet template or copy the custom sheet info from any other design, or from another page in the current design.

See Chapter IV - Schematic Editing for more information.

Custom Sheet Info...

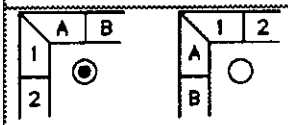
This command displays the Sheet Info palette which allows custom sheet layouts to be created.



The following table summarizes Sheet Info palette controls and options.

See Chapter V - Advanced Schematic Editing for more information on setting custom sheet information.

This Page/All Pages	These buttons determine whether the changes made will be applied to the current page only or to all pages in the current circuit level when the Apply or Done buttons are clicked.
Draw Background Grid	This switch determines whether the gray background grid is drawn in the diagram on the screen. The setting for printed output is determined independently by the "Print Background Grid" item.
Print Background Grid	This switch determines whether the background grid is included in printed or PICT file output.
Draw Default Border	This switch determines whether the default sheet border is drawn around the drawing area on the screen.

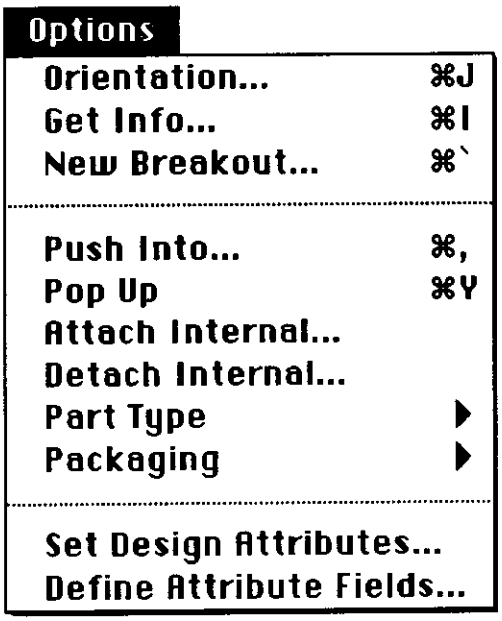
Print Default Border	This switch determines whether the default sheet border is drawn in printed or PICT file output.
Grid Divisions	This number determines how many minor grid divisions are drawn per major division. The default value is 3, entering 1 in effect disables minor grid lines. This has no effect if both Print Background Grid and Draw Background Grid are off.
Border Text	This button displays the standard text style dialog box allowing you to select the text style for the default border. This has no effect if both Print Default Border and Draw Default Border are off.
 (Grid letter/number usage)	These switches determine the usage of letters and numbers for sheet grid references. This setting will affect the position displayed on the tool palette, on Page Connectors, in the Pin List box and in report output.
Auto-Expand	If this switch is on, drawing will be allowed outside the drawing border and the drawing will be automatically expanded to multiples of the sheet size settings. NOTE: This of course means that the drawing will no longer fit on a single sheet and will have to be printed or plotted on multiple sheets.
Use Page Setup	Selecting this option causes the current printer page setup to be used to determine the sheet size. Any changes in page setup (using the Page Setup command in the File menu) will immediately affect the sheet border.
Fixed Sheet Size	Selecting this option causes fixed values to be used for the border size and position, regardless of printer page setup.
Sheet Origin	When this button is selected, clicking in the current schematic page will set the sheet origin point at that position. The sheet origin is the outer edge of the sheet in the corner containing the A1 grid position. The sheet origin can be in any of the 4 corners.

A1 Far Corner	When this button is selected, clicking in the current schematic page will set the far corner (i.e. farthest from the sheet origin) of the A1 grid rectangle. This allows the program to determine the size of the first grid space.
B2 Far Corner	When this button is selected, clicking in the current schematic page will set the far corner (i.e. farthest from the sheet origin) of the B2 grid rectangle. This is used to determine the size of all remaining grid spaces.
Sheet Far Corner	When this button is selected, clicking in the current schematic page will set the far corner of the sheet (i.e. opposite the sheet origin).
Apply	This button applies all settings currently displayed in the Sheet Info palette to the circuit. The palette remains displayed so further changes can be made. THIS CANNOT BE UNDONE!!!
Done	This button applies all settings currently displayed in the Sheet Info palette to the circuit and closes the box. These changes cannot be undone.

Center in Page

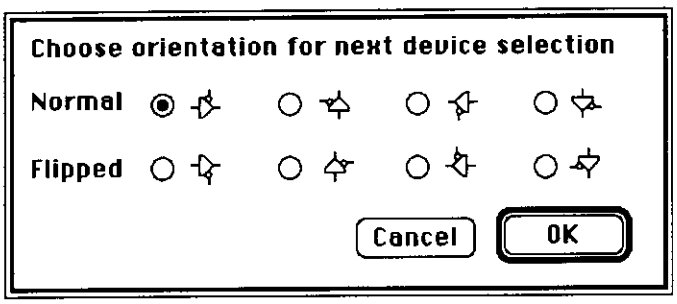
This command moves all items on the current page (except text and picture objects marked as "border" items) so that the circuit objects (taken as a group) are centered in the page. This is intended to assist with situations where a diagram has become lopsided due to modifications, or where a file imported from another system is not correctly situated when it is first read in. If the "Auto-Expand" option is turned on in the "Sheet Info" box, then this command may result in the number of pages required being reduced.

Options Menu



Orientation...

The Orientation command sets the orientation (up, down, left, right, mirrored) that will be used next time a device is created. When this command is selected, the following box is displayed:



The orientation can also be changed by:

- clicking directly on the orientation icon in the tool palette, or
- using the arrow keys on the keyboard.

NOTE: Selecting any menu item cancels the device placement mode, so either of the above methods is preferred since it allows you to change orientation while placing a device or circuit scrap.

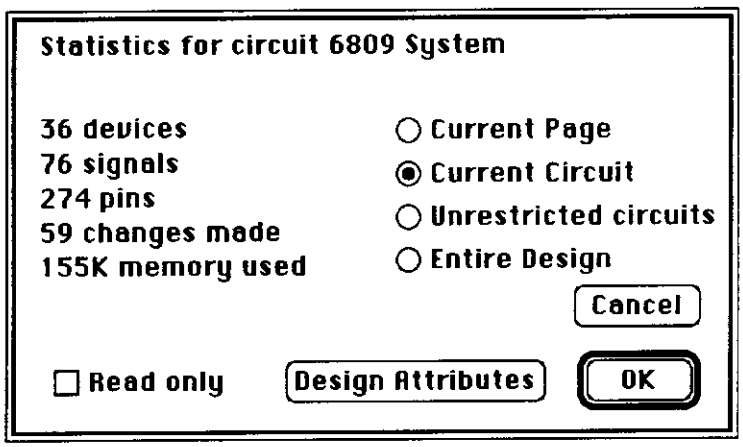
See more information about symbol rotation in Chapter IV - Schematic Editing.

Get Info...

The Get Info command is a general method of viewing and setting parameters and options that are associated with the various types of objects in DesignWorks.

Showing Design and Circuit Info

If no objects are selected in the circuit (i.e. if you have clicked in an empty portion of the diagram) then Get Info will display the following general design information box:



The buttons on the right hand side of this box allow you to chose the scope of the displayed counts:

Current Page	This selection shows information only on objects displayed in the circuit page displayed in the frontmost window.
Current Circuit	This selection shows information on objects on all pages of the current circuit level.
Unlocked Circuits	This selection shows information on objects in the master circuit of the design and all sub-circuits not marked as "Locked" in their respective Device Info boxes.
Entire Design	This selection shows information on all circuits in the design.

The following items of information are shown for the selected scope:

nnn devices	This is a count of devices in the selected scope. Pseudo-devices, such as page connectors and breakouts are not included. In "Current Page" or "Current Circuit" mode, this count includes devices that have sub-circuits. In "Unlocked Circuits" and "Entire Design" mode, this count includes only "bottom-level" devices, i.e. those without a sub-circuit or whose sub-circuit is not being listed.
nnn signals	This is a count of signal nets in the design, including unconnected pins. Signals that pass through more than one hierarchy level are included once for each level that they exist in.
nnn pins	This is a count of device pins, not including pseudo-devices.
nnn changes made	This is a count of editing changes made since the design was created. This is intended to allow comparisons of different versions of the same file.
nnnK memory used	This is a count of the amount of main memory occupied by the selected part of the design, in Kbytes.

Design Attributes - This button displays the standard attribute data box for the current design.

Read Only - This check box allows you to change the "Read Only" status of the current circuit level. This applies only to the current circuit level and is enabled only when the "Current Circuit" button is selected. Note that "Read Only" is not the same as "Locked" in that it allows you to view the circuit but not change it.

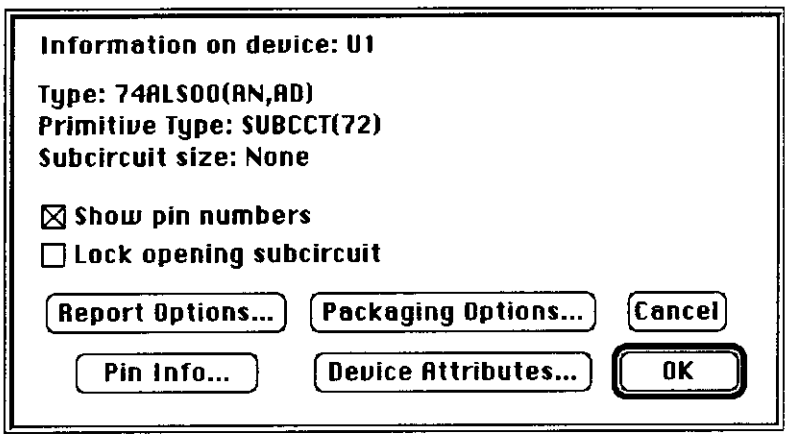
Single Object Get Info

If a single object is selected, Get Info displays a box specific to the object type. To leave any Get Info dialog box, click on the OK button or press the Enter key on the keyboard.

More information on schematic objects is found in Chapter IV - Schematic Editing and Chapter V - Advanced Schematic Editing.

General Device Info Box

When a normal device symbol is selected on the schematic (i.e. not a pseudo-device), then the following information box is displayed:



The following table lists the information and options available in this box.

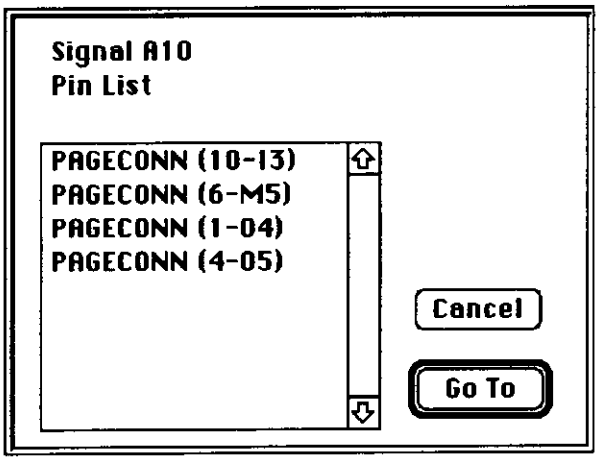
Type	This is the library type name of the device symbol, i.e. the name as it appears in the Parts palette. This is not the same as the Part attribute field, which is normally used as the part name in netlists.

Primitive Type	This is the primitive type of the symbol. For standard types, the name is shown, otherwise the name "Reserved" is shown. The ordinal number of the primitive type value is shown in parentheses.
Subcircuit Size	If the selected device has a sub-circuit, its memory size is shown in Kilobytes.
Show Pin Numbers	This switch allows you to disable the display of pin numbers for the entire device. This is intended for discrete components or others where pin numbers are not normally shown on the diagram.
Lock Opening Subcircuit	This switch allows you to prevent the sub-circuit (if any) of this device block from being opened for editing.
Report Options	This button allows you to select whether this device or its internal circuit will be listed in a netlist.
Pin Info	This button displays the Pin Info box (described below) for the first pin on the device. Buttons on the Pin Info box allow you to sequence through the other device pins.
Packaging Options	This button displays the current packaging level for this device. The meanings of these settings are described in Chapter V - Advanced Schematic Editing.
Device Attributes	This button displays the standard attribute edit box for the device.

NOTE: Clicking Cancel on the Device Info box does not cancel changes that were made in other boxes displayed using Device Info option buttons.

Page Connector Info Box

If a Page Connector device is selected in the schematic, the Get Info command displays the following box:



This is essentially the same box as is displayed for the signal Pin List command, except that only Page Connectors are listed. Following on each item is the page number and grid reference of the item.

To display the selected page connector, either:

- select the item in the list and click the Go To button,

OR

- double-click on the item in the list.

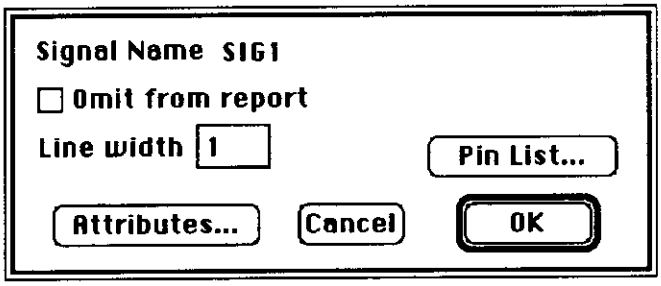
Pseudo-Device Info Box

If a pseudo-device other than a Page Connector is selected in the schematic, the signal Pin List box is displayed for the attached signal. Double-clicking on any item in this list will display the selected pin.

See more information under the Pin List command in the signal pop-up menu, below.

Signal Info Box

Selecting the Get Info command with a signal selected causes the following box to be displayed:

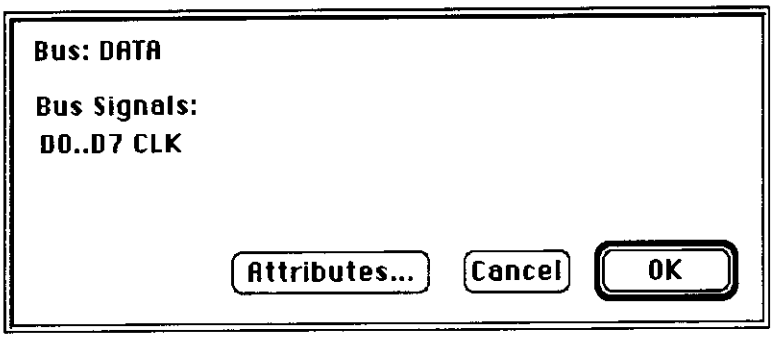


The following table describes the information and options presented in this box:

Omit From Report	This switch controls whether the selected signal is included in any netlist output.
Line Width	The number typed into this box determines the displayed width of the signal. Any number > 1 displays the signal that number of times wider than the normal value. The maximum value is 255.
Pin List	This button displays a list of the real device and pseudo-device pins attached to this signal. Double-clicking on any item in this list will display the selected pin. More information on this box is given under the Pin List command in the signal pop-up menu, below.
Attributes	This button displays the general attribute data entry box for the selected signal. More information on the functions available in this box are given in Chapter VI - Attributes.

Bus Info Box

If a bus line is selected in the schematic, the Get Info command displays the following box:

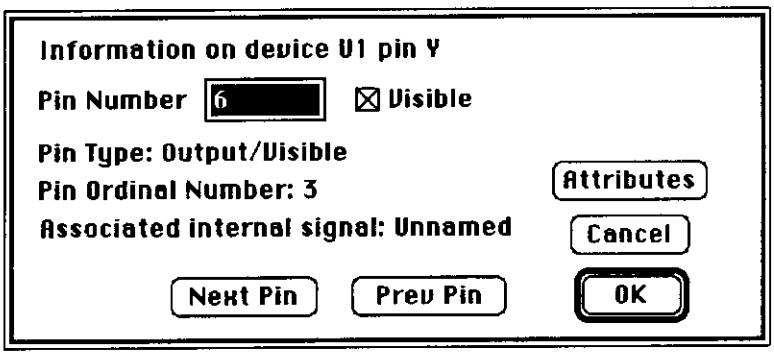


The following table describes the information and options presented in this box:

Bus Signals	This is a list of the signals contained in the bus. This list is determined by the breakouts and bus pins attached to the bus. You cannot directly change this list. See Chapter V - Advanced Schematic Editing for more information.
Attributes	This button displays the general attribute data entry box for the selected bus. NOTE: Except for the Name field, bus attributes are generally not included in any netlist output. More information on the functions available in this box are given in Chapter VI - Attributes.

General Pin Info Box

If the item selected is device pin (non-bus and non-pseudo-device), then the following box is displayed:



The following information and options are available:

Pin Number	This is the physical pin number corresponding to this device pin. This can be empty if desired.
Visible	This check box determines whether the pin number is displayed on the schematic. For some devices, such as discrete components, it may be desirable to have a pin number associated with the pin for netlist purposes without displaying it on the diagram.
Pin Type	This information item gives the function and visible status of the pin.

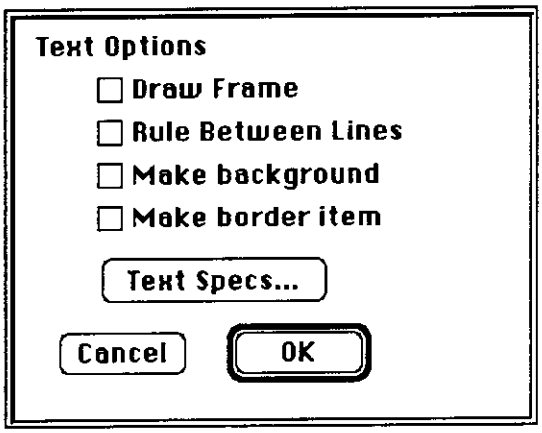
Pin Ordinal Number	This number is the pins ordinal position in the device's pin list, i.e. as viewed in the DevEditor. This can be important in some netlist formats where pin order is critical.
Associated Internal Signal	For sub-circuit devices, this item shows the name of the signal attached to the associated port connector in the internal circuit. For other devices, this will be "None".
Attributes	This button displays the general attribute data entry box for the selected pin.
Prev Pin/ Next Pin	These buttons allow you to move to the next or previous pin (by ordinal number) on the same device, without having to return to the schematic and select the pin.

Pseudo-Device Pin Info Box

If a pseudo-device pin is selected, the Get Info command displays the signal info box for the attached signal.

Text Info Box

If a text object is selected, the Get Info command displays the following box:



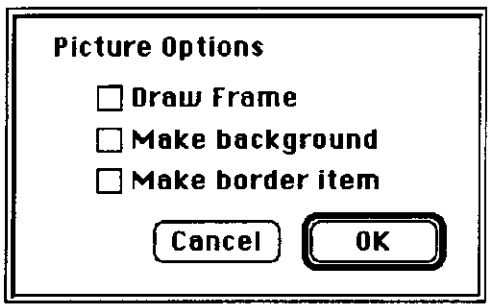
The following table summarizes the options available in this box.

See more information on text objects in Chapter IV - Schematic Editing and Chapter V - Advanced Schematic Editing.

Draw Frame	Turning this switch on causes a frame to be drawn around the text item on the schematic.
Rule Between Lines	Turning this switch on causes a line to be drawn after each row of characters.
Make Background	Turning this switch on makes the selected text block a background item, i.e. it cannot be affected by normal editing operations. In order to select it for editing, hold the Command and Option keys pressed while clicking on it.
Make Border	Turning this switch on marks this item as a . This means that it will be treated as part of the sheet border. I.e. it will be updated by the Import Sheet Info command, copied to new pages as they are created, and unaffected by the Center in Page command.
Text Specs	Clicking this button displays the standard text style dialog. Any changes made in text style affect only the selected item, but they also become the default for future text blocks.

Picture Info Box

If a picture object is selected, the Get Info command displays the following box:



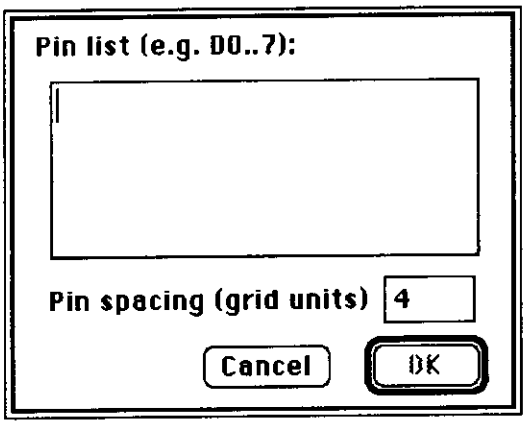
The following table summarizes the options available in this box.

See more information on picture objects for borders and title blocks in Chapter V - Advanced Schematic Editing.

Draw Frame	Turning this switch on causes a frame to be drawn around the picture item on the schematic.
Make Background	Turning this switch on makes the selected picture a background item, i.e. it cannot be affected by normal editing operations. In order to select it for editing, hold the Command and Option keys pressed while clicking on it.
Make Border	Turning this switch on marks this item as a . This means that it will be treated as part of the sheet border. I.e. it will be updated by the Import Sheet Info command, copied to new pages as they are created, and unaffected by the Center in Page command.

New Breakout...

The New Breakout command is used to generate a standard bus breakout symbol for a group of signals. When this command is selected, the following box will be displayed:



The breakout is created by entering a list of signals and the desired pin spacing and clicking the OK button. A flickering image of the breakout will now follow your mouse movements and can be placed and connected just like any other type of device.

See more information on busses and breakouts in Chapter V - Advanced Schematic Editing.

Pin List

Type the list of desired breakout pins into this box. Rules for creating signal lists are as follows:

- blanks or commas can be used to separate individual names in this list, therefore bussed signals cannot have names containing a blank or comma.
- a range of numbered signals can be specified using the following formats:

```
D0..7 or D0..D7
```

is equivalent to

```
D0 D1 D2 D3 D4 D5 D6 D7
```

```
D15..0
```

is equivalent to

```
D15 D14 D13 D12 D11 D10 D9 D8 ... D0
```

```
D15..D00
```

is equivalent to

```
D15 D14 D13 D12 D11 D10 D09 D08 D07 ... D00
```

Note that the ".." format implies that bussed signal names cannot contain periods.

- the signals specified will always appear in the order given in this list from top to bottom in standard orientation. We recommend always specifying numbered signals from lowest numbered to highest, as in the first example above, since this matches the standard library symbols.
- there is no fixed limit on the number of signals in a bus, but we recommend dividing busses up by function (i.e. address, data, control, etc.) for ease of editing.
- the same signal name can appear multiple times in the list, if desired. In this case, these pins will be connected together through the bus.
- any combination of randomly-named signals can be included in the list, as in the following examples:

```
D0..15 AS* UDS* LDS*
```

```
CLK FC0..3 MEMOP BRQ0..2
```

Pin Spacing

The number in the Pin Spacing box will be the spacing between signal pins on the breakout symbol, in grid units. The default value is 4 to match the standard DesignWorks libraries, but any number from 1 to 100 can be entered.

Push Into...

> Notes:
>
> 1) The Schematic tool uses this menu item only for sub-circuit operations. However, other program tools can attach a significance to this menu command for other types of devices having internal data.
>
> 2) If you have used the same device type in multiple places in the design, the Open Internal command creates a temporary type which is distinct from all other usages. When the sub-circuit is closed you will be asked if you wish to update the other devices of the same type.
>
> See "Internal Circuits - Editing in Chapter XXX - Hierarchical Design" for more information.

This command opens the internal circuit of the given device in a separate window. This menu item will be disabled (gray) under any of the following conditions:

- the device is not a SUBCCT (sub-circuit) primitive type
- the device has its "restrict open" switch set in the Device Info box

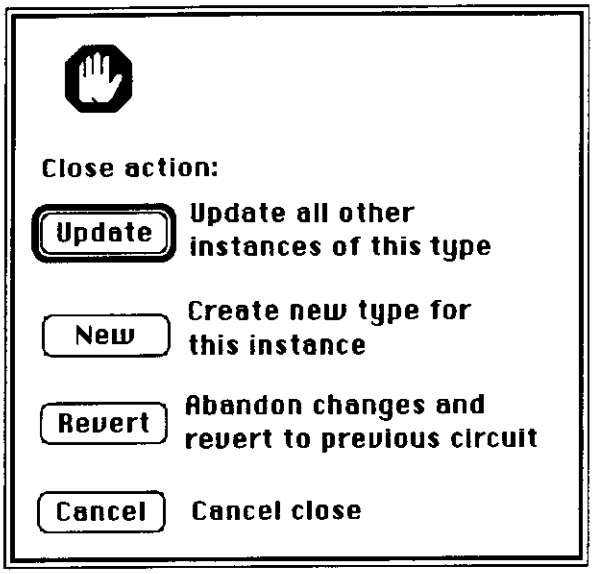
If the selected device has no internal circuit, you will be asked whether you wish to create one. Clicking OK will create a circuit containing only the default port connectors matching the parent device.

Simply double-clicking on a device is a short-cut for the Open Internal command.

Pop Up

This command closes the current sub-circuit and displays the circuit page containing the parent device. If the device being edited was the only one of its type, then the sub-circuit will be closed without comment. If any

changes have been made to the internal circuit that would affect other devices of the same type, then the following box will be displayed:



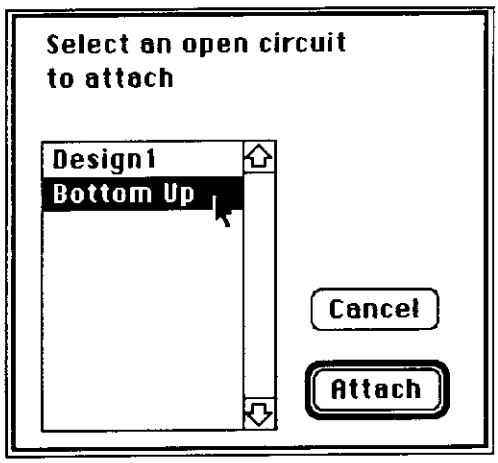
The following close actions are available:

Update	Clicking this button will cause the changes made to be copied to all other devices of the same type. This is the default action and can also be activated by typing the Enter key.
New Type	Clicking this button will make a new type definition for the device that was being edited. It will now be independent of any other devices that used to be the same type.
Revert	All changes made to the internal circuit will be discarded and it will revert to its old structure.
Cancel	Leaves the internal circuit open.

See more information on internal circuits and type definitions in Chapter VII - Hierarchical Design.

Attach Internal...

This command allows you to select an open design to attach as a sub-circuit to the selected device. When this command is selected, the following box will appear:



WARNING: This operation cannot be Undone!!!

Clicking Attach on the Attach Internal box will cause the following actions to be taken:

1) If the current design (i.e. the one containing the parent device) contains other devices of the same type as the selected device, then a separate, temporary type will be created for the selected device, as is done with the Push Into command, above.

2) The attribute definition table in the selected internal circuit is compared with the table in the master design. If the Attach operation would result in new fields being defined in the master design's table, then you will be prompted for confirmation. If you click OK, then new fields are merged into the master design's table.

3) Attribute values associated with the internal circuit are merged into the master design. If a given field already has a value in the master design, the value in the internal circuit is discarded.

4) The logical linkage between the selected device and the new internal circuit is completed. If any mismatch is detected between the port connectors defined in the internal circuit and the pins on the parent device, you will be warned.

5) The title of the internal circuit is updated to reflect its hierarchical position in the master design.

6) Auto-packaging, if enabled, is disabled since this operation potentially invalidates the design's package table.

7) The newly-attached internal circuit's window is brought to the front. It is now considered to be an internal circuit that has been opened for editing and modified. When you close the internal circuit, you will be asked if you wish to update other devices of the same type.

For more detailed information on the Attach Internals operation, see Chapter VII - Hierarchical Design.

Detach Internal...

This command makes the currently displayed sub-circuit into a separate design and redefines the parent device as having no internal circuit.

WARNINGS:

1) This operation permanently removes the sub-circuit from the selected device and all other devices of the same type in the selected design. If you do not wish to update other devices in the design, use the Make Unique command to isolate the selected device first.

2) The Detach operation cannot be Undone!!!

In particular, Detach Internal performs the following operations on the sub-circuit displayed in the frontmost window:

1) The circuit is unlinked from its parent device, making it into a separate design.

2) The title of the sub-circuit is set to a default "Designxxx" name.

3) The internal circuits of all other devices of the same type in the design are removed.

4) Auto-packaging, if enabled, is disabled since this operation potentially invalidates the design's package table.

For more detailed information on the Detach Internal operation, see Chapter VII - Hierarchical Design.

Part Type

This item displays a sub-menu of commands relating to device types. See the description of these command below.

Packaging

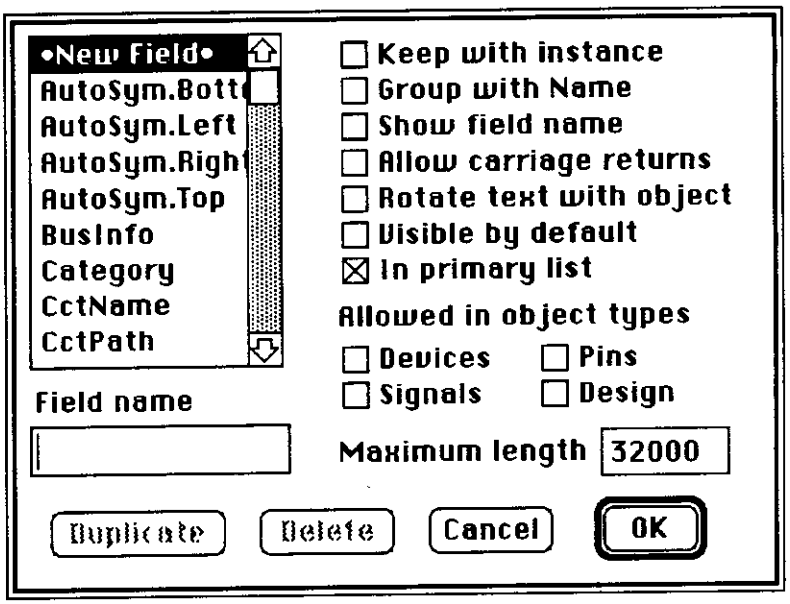
This item displays a sub-menu of commands relating to device packaging. See the description of these command below.

Define Attribute Fields...

This command allows you to add or modify attribute field definitions for the current design. A short summary is given here.

See Chapter VI - Attributes for more information.

When this command is selected, the following box is displayed:



Note the following characteristics of these settings:

- Changing the values of any of the option switches for an existing field do not affect any values already in the design. Only new data will be affected. See Chapter VI - Attributes for tips on using the Duplicate function to update the way existing data is displayed.
- The name and option settings for most predefined fields are fixed and will be disabled in this box.
- Regardless of changes made in this box, no design data is updated until you click the OK button.

• At least one of the "Allowed in Object Types:" switches must be selected.

Option	Meaning
Keep with Instance	This option applies only in Physical Hierarchy mode. When this switch is checked, the selected attribute field may take on a different value for each instance of a device, pin or signal.
Group with Name	When this switch is checked, any value entered for this field will be displayed beneath the Name field by default. When the Name is moved, this field will also be moved.
Show Field Name	When this switch is checked, the field name will be shown with the value on the schematic.
Allow Carriage Returns	If this switch is checked, then the Return key may be used to enter Carriage Return characters into this field.
Rotate Text With Object	If this box is checked, displayed values will be rotated if the associated object is rotated on the schematic. Otherwise, when the object is rotated, the text will move to an appropriate new position but be drawn in the normal orientation.
Visible by Default	If this box is checked, whenever a value is entered for this field it will be displayed on the schematic in a default position. If the field is found in the default attributes for a device, the field will be displayed automatically on the diagram when the device is placed.
In Primary List	If this box is checked, the field will be considered a "primary" field and will be displayed in all attribute entry boxes when "Primary Only" is selected.
Allowed in Object Types	This group of check boxes allows you to select which types of schematic objects may have this field associated with them. You can check any one or more boxes.

Maximum Length	You can specify a maximum length (in characters) for the field data.
Field Name	This box allows you to enter the name of a new field or rename an existing one. If you rename a field so that it has the same name as another existing field, this becomes a Merge operation. You will be prompted to confirm this operation before proceeding.
Duplicate	This button allows you to create a new field with the same settings and (optionally) data contents as an existing one. You will be prompted to confirm the data duplication option before proceeding.
Delete	This button allows you to permanently remove a field definition and all associated data from the design. THIS CANNOT BE UNDONE!!!

Set Design Attributes

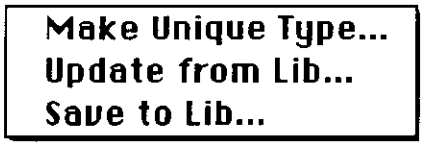
This command displays the standard attribute data entry box allowing you to enter attribute data for the design itself. Design attributes can be used to create variables in text blocks on the schematic and to insert global data in netlist output.

See Chapter XI - Report Generation for more information on design attributes in netlists.

See Chapter V - Advanced Schematic Editing for more information on text variables.

See Chapter VI - Attributes for more information on attribute data entry.

Part Type Sub-Menu



WARNING: The commands in the Part Type sub-menu can have major hidden effects on the structure of a hierarchical design. Do not use them without a clear understanding of what they do!

Make Unique Type...

This command makes the selected device into a unique type. In other words, even if other devices in the same design were originally drawn from the same library, this one will now be considered to be unique. This is primarily intended for hierarchical designs. This allows the internal circuit for the selected device to be modified or deleted without affecting other similar devices.

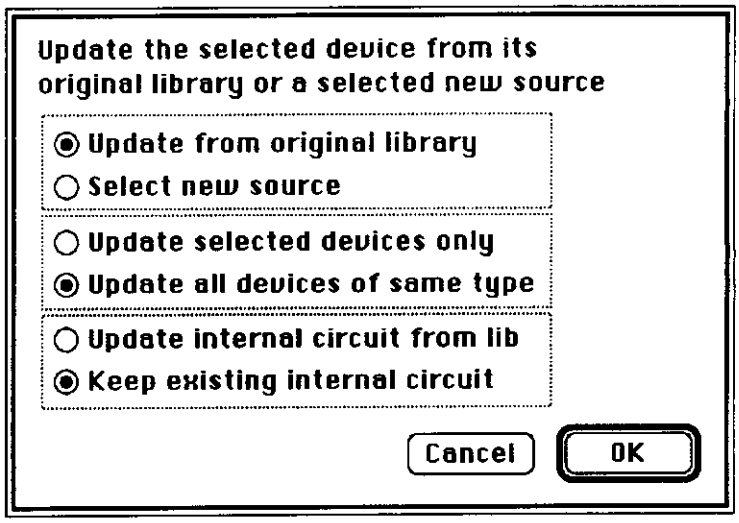
As an example, you could create a standard template for a Programmable Logic Device in a library, complete with internal simulation circuit. Several of these devices could be placed in the design, even though they will eventually have different internal programming. The Make Unique Type command would then be used to isolate them so that their internal circuits could be separately updated.

Update from Lib...

This command updates the definition of the selected device from its original source library.

WARNING: Depending upon the options selected, this command may replace the entire internal circuit of the selected device. THIS CANNOT BE UNDONE!

Selecting Update from Lib causes the following box to be displayed:

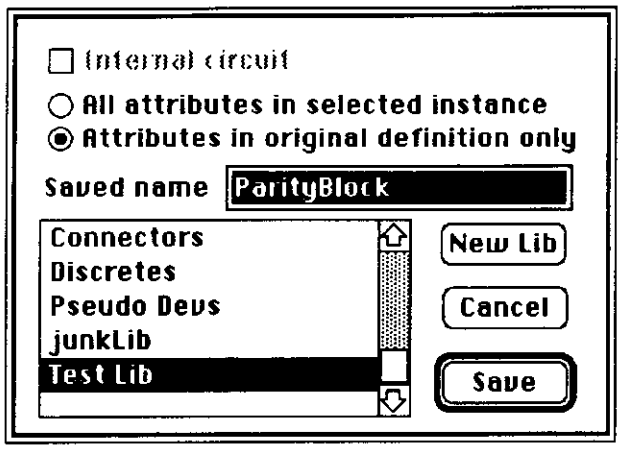

Update the selected device from its original library or a selected new source
Update from original library
Select new source
Update selected devices only
Update all devices of same type
Update internal circuit from lib
Keep existing internal circuit
Cancel
OK

The following table summarizes the options available.

Update from Original Library	The original library and part name will be used to locate a new part definition. The LibName, LibPath, LibDate and LibType attribute fields are used.
Select New Source	A box is displayed allowing you to select any open library and part as the source for the update operation.
Update Selected Device Only	Only the selected device will be updated. NOTE: This in effect redefines the device to be different from others that were originally derived from the same library part.
Update All Devices of the Same Type	All devices of the same type will be updated.
Update Internal Circuit from Lib	If the selected device has an internal circuit it is removed. If the new library part definition has an internal circuit as new one is created.
Keep Existing Internal Circuit	Any existing internal circuit is kept and only the symbol is updated.

Save to Lib...

This command saves a type definition for the selected device to a library. The following box will be displayed:



The following table summarizes the options available.

Internal Circuit	If this box is checked, the internal circuit attached to the selected device will be saved with the part definition.
All attributes in selected instance	If this option is selected, all the attribute values associated with the selected device will be made part of the saved library part.
Attributes in original definition only	If this option is selected, only attributes that were originally defined for the library part will be saved.
Saved Name	The part name under which the new library entry will be saved.
New Lib	This button will display a standard file save box allowing you to create a new, empty library.

Packaging Sub-Menu

The menu items in the Packaging sub-menu are used to control the automatic gate package assignment feature. The Packager can make name and unit assignments either interactively (i.e. as schematic changes are made) or in batch mode (i.e. only on command).

Following is a short description of the packaging commands.

See Chapter V - Advanced Schematic Editing for more information.

Auto Packaging Enabled

This command turns automatic device packaging on and off. When auto-packaging is enabled (i.e. the menu item is checked), the Packager automatically assigns a name and unit number to each device symbol that is placed in the design. When disabled, the device attribute fields associated with packaging are left at their default values for new devices. Packaging can then be assigned manually by the user, or as a batch using the Repackage Design command.

NOTE: Auto-packaging cannot be enabled if the design contains packaging errors such as duplicate gate assignments or invalid packaging attribute data. If the package table is out of date when you attempt to enable packaging, the Packager will first prompt for permission to rescan the design.

Default Prefix...

This command sets the default package name prefix. This prefix is used by the Packager to generate a device name if the Name.Prefix field is not specified in the device. This value is stored in the design's PkgPrefix field.

Repackage Design...

This command forces the entire design to be repackaged, i.e. new name and unit assignments are made for all devices in the design not marked as "Locked" or "Ignore". This should be done if auto-packaging has been off while manual editing changes were made, or if parameter changes have been made that would affect packaging.

The Repackage operation performs the following steps:

- Clear the design's package table.
- Scan all devices in the design, placing all devices marked as "Lock and Check" or "Lock and Don't Check" in the package table. This ensures that these names will not be used for automatic unit assignment. Devices marked "Ignore" are not placed in the table, meaning that their names could be used in automatic unit assignment. Any errors detected in the package assignment of devices marked "Lock and Check" are reported.
- Sort all devices in the design by hierarchy level, then by page, then by horizontal grid position, then by vertical grid position.
- In the sorted order defined above, assign a new package name and unit to all devices not marked "Lock" or "Ignore".

Rescan Design...

This command forces the Packager to rebuild the design's package table from the current state of all device attribute fields. This can be used in the following circumstances:

- To check for errors or obtain a Free Unit Report after manual packaging changes have been made.
- After back annotation or any other external process that may have changed package assignments.
- After changing any of the attribute fields that may affect packaging.

If any packaging errors are encountered (e.g. duplicate gate assignment or invalid attribute fields), they will be reported at the end of this process and the design's package table will be considered invalid. No device attribute fields are affected by this command.

Free Unit Report...

This command generates a text report file listing free device units available in the current design.

Reassign Device Names...

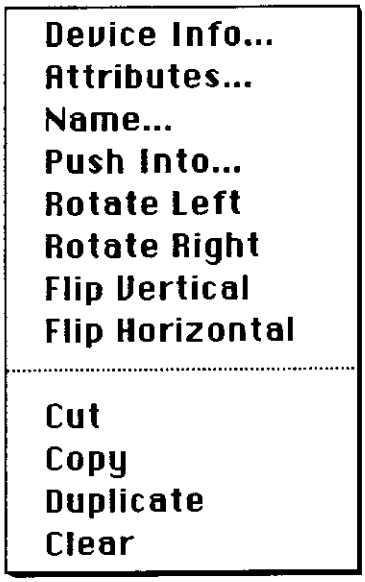
This command is a more restricted version of the Repackage Design command and can be used in any hierarchy mode. It assigns a new name to all devices in the current circuit that are either unnamed or have a default name. It is intended as a quick means of tidying up automatically-assigned names when a circuit is created or edited.

NOTE: If the design is in "Flat" hierarchy mode (the default) then this command will override the gate package assignments made by the Packager. In other hierarchy modes, the Packager stores the package assignments in the "InstName" field, which is unaffected by this command.

Note the following rules:

- Only the current circuit is affected. Other circuit levels higher or lower in the hierarchy are unaffected.
- Gate packaging information is ignored, i.e. every device symbol is assigned a unique name regardless of package availability or "locked" status.
- The "Unit" attribute field is unaffected.
- Only the "Name" field is changed, regardless of hierarchy mode.
- Names are assigned in order sorted first down the columns of the references grid, then across the page, then through subsequent pages.
- Names are assigned using the device's prefix field (i.e. the one named in the design's "PrefixField" field) or the design's "DevPrefix" field.
- Names that have been edited by the user are no longer considered "default" names and will not be changed.

Device Pop-Up Menu



The device pop-up menu is displayed by holding the ⌘ key pressed while clicking and holding on any device in the current circuit.

Device Info...

This command displays the general device information box. This is equivalent to selecting the device and using the Get Info command. See more information under the Get Info command above.

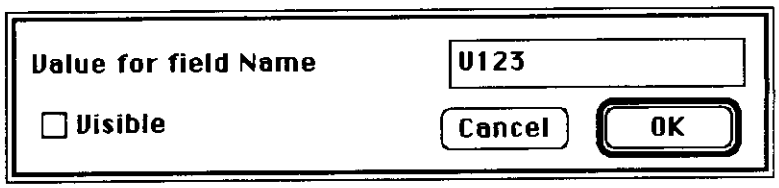
Attributes...

This displays the standard Attribute Data box allowing you to enter or edit attribute data for the selected device.

See Chapter VI - Attributes for more information.

Name...

This command displays a simple edit box allowing you to enter or edit the device name. This provides a simpler method of editing only the name, as an alternative to using the Attributes command above.



Push Into...

This command is equivalent to the Push Into command in the Options menu. See that entry for more details.

Flip and Rotate Commands

These four commands are equivalent to deleting the selected device and replacing it in the selected new orientation. If the new rotation causes any device pins to touch adjacent signal lines, connections will be made unless the Option key is held.

The Clipboard is not affected by these commands.

See more information on device rotation in Chapter V - Advanced Schematic Editing.

Cut

This is equivalent to selecting the Cut command in the Edit menu while this device is selected. The device is copied to the Clipboard and removed from the circuit.

Copy

This is equivalent to selecting the Copy command in the Edit menu while this device is selected. The device is copied to the Clipboard.

Duplicate

This is equivalent to selecting the Duplicate command in the Edit menu while this device is selected. The given device is duplicated and the program enters Paste mode immediately.

The Clipboard is not affected.

Clear

This is equivalent to selecting the Clear command in the Edit menu. The device is deleted from the circuit.

The Clipboard is not affected.

Attribute Value List Sub-menus

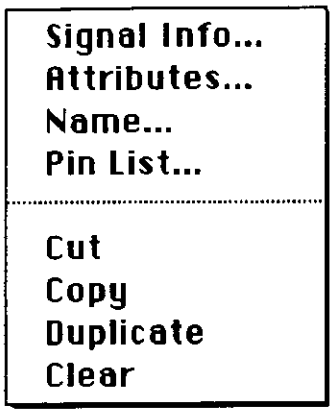
The device pop-up menu may contain one or more sub-menus as its last items. These sub-menus allow you to select from a list of possible values for a device attribute field. Selecting an item in one of these lists changes the value in the associated attribute field.

Most of the standard libraries provided with DesignWorks contain fields defining the part code, the package type and the gate unit (e.g. a, b, c, etc.). When one of these fields has more than one possible value, the list is displayed in one of these menus. For example, if the sub-menu "Unit" appears, this means there are multiple gate units available in this device type. Selecting one of the units in the Unit sub-menu causes the Unit attribute field to be changed for that device and the pin numbers to updated accordingly.

Users can add their own value list sub-menus for any attribute field where it is convenient to select one of a number of possible values. Any field defined for the design, called for example "Value", can have another field associated with ".List" appended to the name, so in this case "Value.List". The "Value.List" field should contain text items separated by commas. Each one of these items will appear on a separate line in the "Value" sub-menu in the device pop-up.

See Chapter VI - Attributes for more information.

Signal Pop-Up Menu



The signal pop-up menu is displayed by holding the ⌘ key pressed while clicking on any signal line.

Signal Info...

This command displays the general signal information box. This is equivalent to selecting the signal and selecting the Get Info command in the Options menu. See more information on this command above.

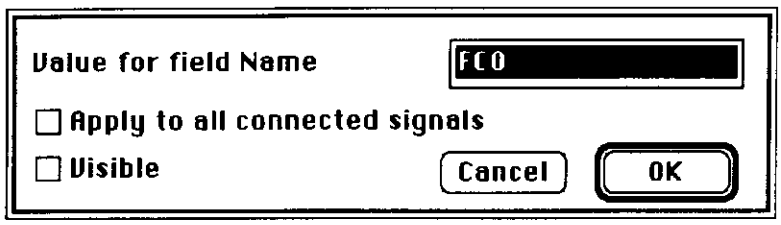
Attributes...

This command displays the general attribute data entry box for the selected signal.

> NOTE: For name changes, it is best to use the Name command described below since it provides more options for applying the name change.

Name...

This command displays a name edit box for the selected signal. The following box is displayed offering you several options on how the name changed is to be applied:



Apply to All Connected Signals

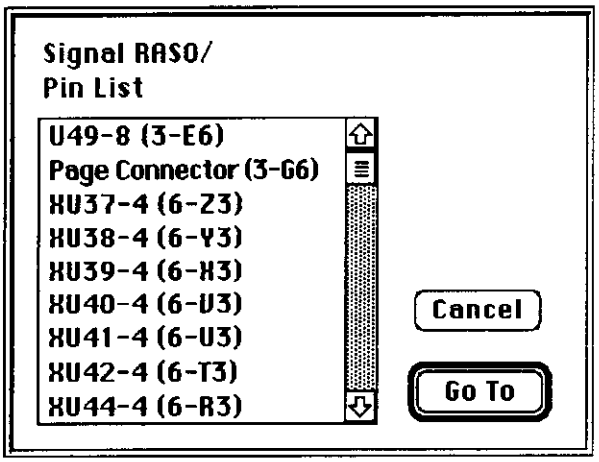
This option allows you to choose whether the name change applies only to the selected signal segment (i.e. thereby breaking its connection with other like-named signals), or to all interconnected signal segments.

Visible

This option allows you to choose whether the name entered should be displayed on the schematic. If the name was already visible and you uncheck this box, it will be removed from the schematic. In this case, the name will still be associated with the signal as an invisible attribute. If the name was not previously visible and you check this box, it will be displayed somewhere adjacent to one of the signal line segments.

Pin List...

This command displays a list of all device pin connections comprising the selected signal. Only pins in the current circuit are listed; pin connections in other hierarchy levels are not shown.



Selecting any item in the pin list and clicking on the Go To button will cause the appropriate page to be opened (if necessary) and that pin to be displayed in the center of the circuit window. If the selected pin is invisible (e.g. a bus internal pin) the window will be centered on the parent bus pin.

For convenience in navigating a schematic, connections to pseudo-devices such as bus breakouts and ground symbols are also shown in this list. In this sense, the list is not a logical "netlist", but rather a method for checking connectivity and navigating the diagram.

Each item in the pin list is formatted as follows:

For normal devices: *device-pin (page ref)*

For pseudo-devices: *device type (page ref)*

The elements of this format are as follows:

device	If the device is named, this will be its name, otherwise it will be a "#" mark plus the type name.
pin	If the pin has a pin number, this will be the pin number, otherwise a "#" and the pin name.
page ref	This will be a reference to the page number and grid position, formatted as specified in the Design Preferences box.

Cut

This is equivalent to selecting the Cut command in the Edit menu while this signal is selected. The signal is copied to the Clipboard and removed from the circuit.

Copy

This is equivalent to selecting the Copy command in the Edit menu while this signal is selected. The signal is copied to the Clipboard.

Duplicate

This is equivalent to selecting the Duplicate command in the Edit menu while this signal is selected. The given signal is duplicated and the program enters Paste mode immediately.

The Clipboard is not affected.

Clear

This is equivalent to selecting the Clear command in the Edit menu. The signal is deleted from the circuit.

Attribute Value List Sub-menus

The signal pop-up menu may contain one or more sub-menus as its last items. These sub-menus allow you to select from a list of possible values for a signal attribute field. Selecting an item in one of these lists changes the value in the associated attribute field.

None of the predefined fields for signals have value lists associated with them. Therefore, a value list sub-menu will only appear here if user-defined value list fields have been created.

See Chapter VI - Attributes for more information.

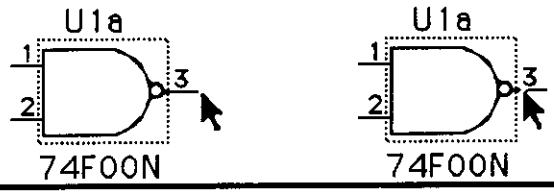
Pin Pop-up Menu

The pin pop-up menu is displayed by holding the ⌘ key pressed while clicking on any device pin.

NOTE: Since an unconnected device pin is both a pin and a signal you determine whether you get the pin or signal pop-up menu as follows:

- clicking on the pin in the last 1/4 of the pin length away from the device will display the signal menu.
- clicking on the pin close to the device symbol will display the pin menu.

Selecting the Signal **Selecting the Pin**



Pin Info...

This command displays a standard Pin Info box showing the name, function and internal circuit association of the pin, as well as allowing you to edit the pin's attributes.

The Next and Previous buttons on this box can be used to view and edit other pins on the same device without having to return to the schematic and select them individually.

This command is equivalent to selecting the pin and choosing the Get Info command in the Options menu.

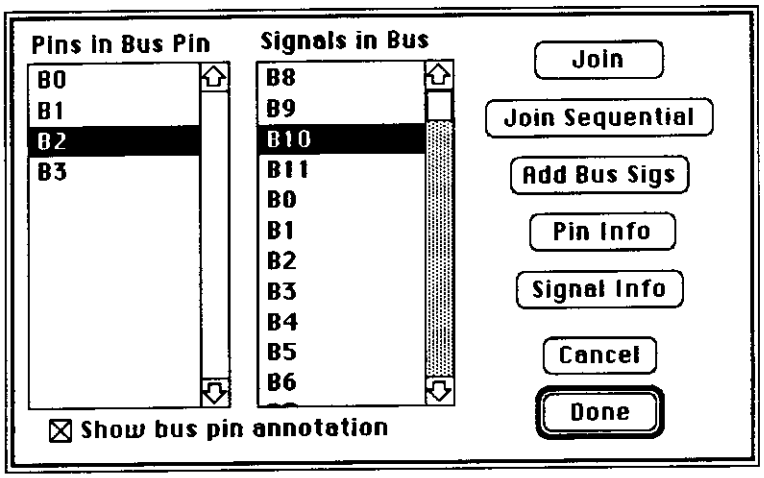
Attributes...

This command displays the standard attribute edit box for fields associated with the selected pin.

The Next and Previous buttons on this box can be used to view and edit other pins on the same device without having to return to the schematic and select them individually.

Bus Pin Info...

This menu item will be enabled only when a bus pin on a device is selected. It allows the association between the bus internal pins on the device and the signals in the bus to be changed. The following box will be displayed:



The left-hand list shows the names of the pins contained in the selected bus pin. The right-hand list shows all the signals in the attached bus. For each pin in the pin list, the signal in the same row in the signal list is the one attached to it. Signals in the signal list beyond the end of the pin list are not connected in this bus pin.

Changing Signal Connections

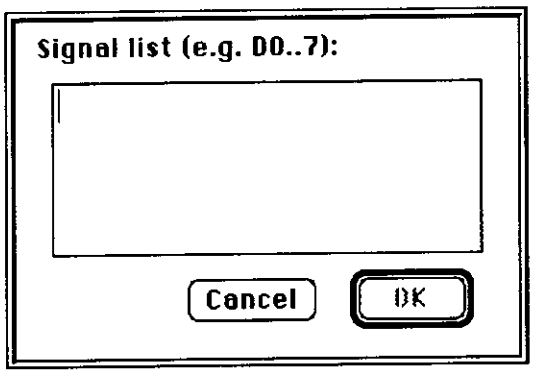
Two buttons are provided to change the association between pins and signals. The Join button causes the selected pin in the pin list to be joined to the selected signal in the signal list. If the selected signal is already attached to another pin in the list, then the signals will be swapped (i.e. a signal can only connect to one pin and vice versa). The signal list will be updated to show the new relationship.

The Join Sequential button provides a quick method of joining multiple numbered pins and signals. The selected pin is joined to the selected signal, as with Join, above. If the signal and pin names both have a numeric part, both numbers are incremented and the corresponding signal and pin are joined. This process is repeated until either the signal or pin name is not found in the list.

For example, given the lists appearing in the above picture, if pin B0 and signal B4 are selected, then Join Selected will join B0-B4, B1-B5, B2-B6 and B3-B7. Since there are no more numbered pins, the process would stop. Note that although the signal and pin names are the same in this example, this is not a requirement.

Add Bus Sigs

The Add Bus Sigs button allows you to add signals to the signal list so that they can then be joined to device pins. Clicking this button displays the following box:



A list of signals can be typed into this box using the same format as the New Breakout command. Following are examples of allowable formats:

```
D0..7
D0..15 AS* UDS* LDS*
CLK FC0..3 MEMOP BRQ0..2
```

The order of entry will affect the order that the signals appear in the list, but is otherwise not significant. For a complete description of the rules of this format, see the New Breakout command elsewhere in this chapter.

NOTE: These signals are only added temporarily. When you close the Bus Pin Info box, all signals that are not connected to any pin are removed from the bus.

Pin Info Button

The Pin Info button brings up the standard Pin Info box for the pin selected in the pin list. See the Get Info command for more information.

Signal Info Button

The Signal Info button brings up the standard signal info box for the signal selected in the signal list. See the Get Info command for more information.

Show Bus Pin Annotation

If this option is enabled, a list of the connections made in the bus pin will be displayed adjacent to the pin. This is done by creating a value in the BusInfo attribute field for the selected bus pin. This value can be edited manually, if desired, but will be updated automatically each time this box is displayed. The format of the signal list is the format used by the Add Bus Sigs option, above.

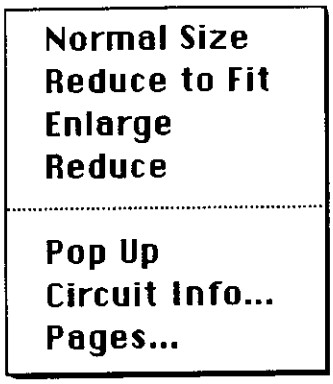
Attribute Value List Sub-menus

The pin pop-up menu may contain one or more sub-menus as its last items. These sub-menus allow you to select from a list of possible values for a pin attribute field. Selecting an item in one of these lists changes the value in the associated attribute field.

None of the predefined fields for pins have value lists associated with them. Therefore, a value list sub-menu will only appear here if user-defined value list fields have been created.

See Chapter VI - Attributes for more information.

Circuit Pop-up Menu



Normal Size/Reduce to Fit/Enlarge/Reduce

These menus items control the zoom scale factor and function exactly as the same-named items in the Drawing menu except that they attempt to zoom in around the area of the mouse click.

Pop Up

This command is equivalent to the same-named item in the Options menu. It closes the current circuit and displays the circuit containing the parent device. It will be disabled if the current circuit is the top level in the design.

Circuit Info

This command is equivalent to the Get Info command in the Options menu while no items are selected in the current circuit. It displays the circuit info box.

Pages...

This command is exactly the same as the Pages... command in the Drawing menu. It displays the Page box allowing you to create, delete, title and display circuit pages.

Attribute Pop-up Menu

When any visible attribute data item on the schematic is clicked in with the ⌘ key pressed, the Attribute pop-up menu will appear.

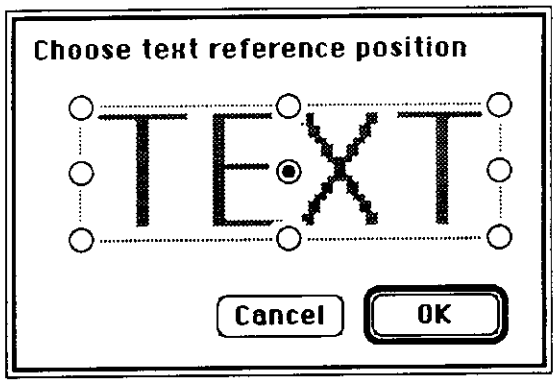
NOTE: Even though clicking on an attribute item selects and highlights the object associated with it, the commands in this menu affect only the attribute field that was clicked on.

Edit...

This command opens a text box allowing you to edit the contents of the selected field. All locations where this field is displayed on the schematic will be updated when the OK button is clicked.

Justification

This command allows you to change the vertical and horizontal justification used in the positioning the attribute text on the diagram. When this command is selected, the following box will be displayed:



The selected point on the text is considered to be the reference point for the given attribute block. This point will be kept fixed if any field value or text style changes cause the box to be resized.

For more information on attribute text justification, see Chapter VI - Attributes.

Hide

This command causes the visible attribute text that was clicked on to be removed from the schematic without removing the value from the field. I.e. if you select the associated object and open the attributes edit box, the same value will still be present. If the field was displayed in more than one place, only the one clicked on will be removed.

Clear

This command causes the value for this field to be set to null and all visible occurrences of it on the schematic to be removed.

Duplicate

This command creates another visible occurrence of the same attribute field. This text can then be dragged or rotated to any desired position on the schematic.

Rotate Left/Rotate Right

These two commands cause the single visible attribute text item that was clicked on to be rotated in the given direction.

Show Field Name

This command allows you to display the field name with the value on the schematic. When this item is checked, the display will be in the form "fieldName=value". Selecting this command again will cause the display to revert to the normal value display. This command applies only to the selected field on the selected object.

This page is intentionally left blank.

Chapter X - Device Symbol Editing

Introduction

The Device Editor is a tool used to create device symbols or general graphics (e.g. title blocks or mechanical drawings) for use on DesignWorks schematics. It provides a complete, object-oriented drawing environment with standard drawing tools, as well as specific functions tailored for symbol creation.

Installation

The DevEditor is a code module that works with DesignWorks using the MEDA interface. In order to be located when DesignWorks starts up, it must be in the same folder with the DesignWorks program itself, or in the folder specified in the TOOLFOLDER statement in the DesignWorks setup file, usually the "Tools" folder.

Memory Usage

The DevEditor tool occupies a significant amount of memory space (about 230K) regardless of whether or not it is being used. If you are operating on a system with a minimal amount of memory, it may be desirable to keep the DevEditor tool in another folder where it will not be found and loaded by DesignWorks . When it is actually needed, it can be moved temporarily into the DesignWorks folder.

Starting the DevEditor

To Create a New Part

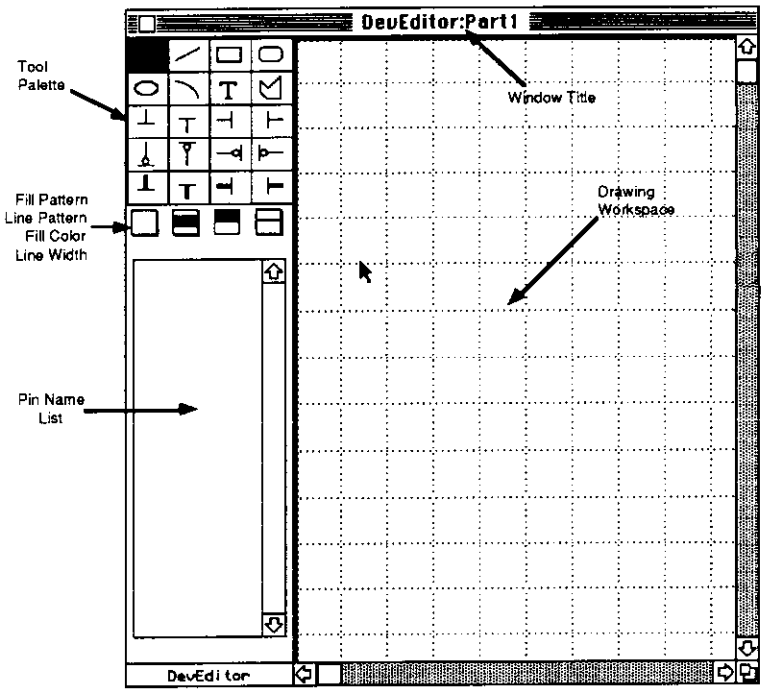
Select the name DevEditor in the Tools menu. This will open an empty graphics window, allowing you to create a new device type from scratch. If there is already an open DevEditor window this will bring it to the front.

Editing Existing Part

1) Select the part to be edited in the parts palette

2) Choose the Edit Part command either in the Libraries submenu in the File menu or in the Parts palette pop-up menu.

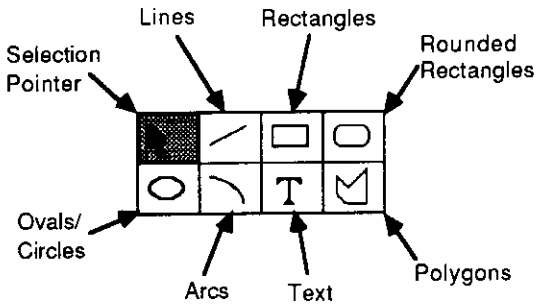
The DevEditor Window



Window Title	"DevEditor:" followed by the name of the part being edited.
Tool Palette	Used to select a graphical editing or pin placement tool. The current tool is displayed selected.
Fill Pattern	Pop-up palette. Graphic objects are filled with this pattern.
Line Pattern	Pop-up palette. Graphic objects have their edges drawn with this pattern.
Fill Color	Split pop-up palette. The top half of the pop-up is used to set the foreground color of a graphic object, the bottom half of the palette sets the background color for an object.
Line Width	Pop-up palette. Used to select the width of lines.

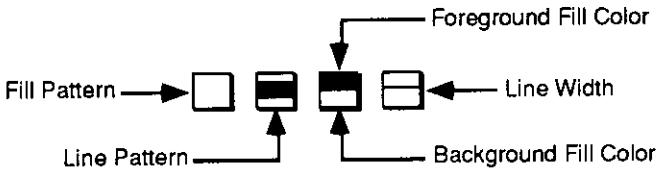
Pin Name List	Scrolling list of names for all of the pins which exists. A pin exists as long as it is in the pin name list. It need not have a graphic representation to still be valid.
Drawing Workspace	This area is used to construct the graphical appearance of the part.

Drawing Tools



The 8 items in the top half of the tool palette represent standard Macintosh drawing tools.

Graphic Palettes

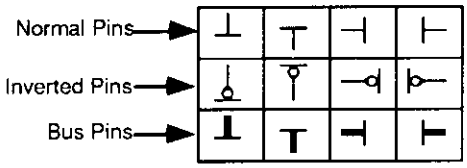


The fill pattern, line pattern, fill color, and line width palettes change how the selected graphical objects are drawn. If there are selected objects when a palette is changed then the selected objects will be given the characteristic selected. If there are no graphical objects selected when a change is made to a palette, the default setting for the palette is changed. Graphical objects created in the future will use the new default setting.

Fill Pattern Pop-up	The fill pattern palette ❑ determines the pattern used to fill the inside of ovals, rectangles and polygons. The "N" item refers to "No fill", which is the default.

Line Pattern Pop-up	The line pattern palette sets the pattern used for all outer framing lines on objects.
Fill Color Pop-up	The color palette is split into upper and lower halves which determine the colors used for the "foreground" (i.e. "black" areas of lines, text and fill patterns) and "background" (i.e. "white" areas of text and fill patterns).
Line Width Pop-up	The line width palette determines the width of lines and object frames.

Part Pin Tools



The top row of part pin tools are used to create pins on the part in any of the four orientations.

The second row of pins are provided as a convenience to create pins with an inversion bubble. A pin created with the inverted pin tools has no special characteristics, these are only provided as a graphical convenience.

The third row of pins are used to place bus pins on the part in any of the four orientations. Bus pins allow buses to be connected to the part. A bus pin's functionality is determined by the internal pin names it contains. See the Add Pins command below for information on creating bus internal pins.

When placing pins, a graphical pin is associated with a name in the pin list. The association is made by applying the following rules and using the first one that matches.

1) Associate a selected pin name in the pin list which is unplaced and is of the same type. I.e.: normal and inverted pins can't be associated with bus pin names or internal bus pin names.

2) Associate an unplaced pin name of the correct type which follows the first selected pin name. Pin names are examined in a cyclic order so if the bottom of the pin name list is reached the search continues from the beginning.

3) If no association is made then a new pin name of the correct type is created and added to the bottom of the list.

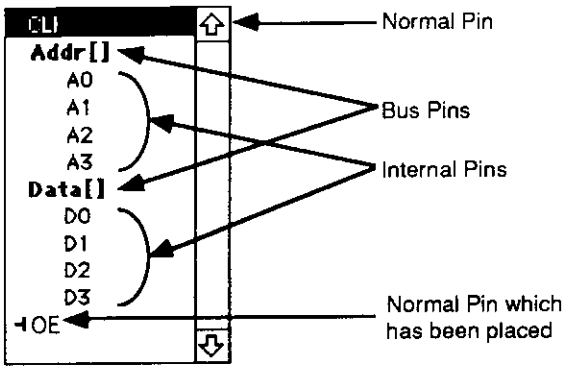
Pin Name List

The pin name list box contains a scrollable list of the pin names associated with this device. This list is derived from the following sources:

- if the symbol was opened from an existing part, then the initial name list will be the pin names associated with the part.
- if a pin tool is clicked in the drawing workspace, a new name may be added with the form "PINxx" where xx is a sequential number. This is only done if no unplaced pin name could be used.
- if an open DesignWorks circuit is selected as an internal circuit, or a saved DesignWorks circuit is selected as an external circuit, then the names of the port connectors in the circuit will be merged with the names in the pin name list.
- the Add Pins dialog can be used to create a list of pins to be merged with the pin names in the pin name list.
- the Auto Create Symbol command can be used to specify all the pin names and pin numbers for a device symbol.

The pin name that appears in the pin name list does not need to be the same as the graphical annotation which appears next to a graphical pin in the part's symbol. Some operations performed by the DevEditor - Auto Symbol Create - assume a correspondence but this is only done as a convenience. The result is that changing a pin name in the pin name list will never cause the annotation next to a graphical pin to automatically change.

Any pin name that has been placed (i.e. it has a corresponding graphical pin object which appears as part the symbol) will have a ⊣ mark beside it. Bus pins are marked in the list with "[ ]" following their name and the are displayed in bold. Internal pin names (i.e.: pin names which define the contents of a bus) are indented when displayed.



If a placed pin name (i.e.: it has an associated graphical pin, and a "⊣" mark next to its name) is selected then both the pin name and the graphical name will be selected. The opposite is also true. If a graphical pin is selected so will its pin name. If a pin name is selected which has not been placed, then only the name is selected. Internal pin names never have graphical associations, therefore selecting an internal pin name never selects a graphical pin.

The relationship between pin names and graphical pins is not symmetrical. Every graphical pin must have a pin name, but every pin name may not have an associated graphical pin. This can lead to some surprises. For example,

- selecting all graphical pins in the drawing workspace may not select all pin names, unplaced and internal pin names will not be selected.
- deselecting all graphical pins in the drawing workspace may not deselect all pin names. Additional unplaced or internal names may have been initially selected.

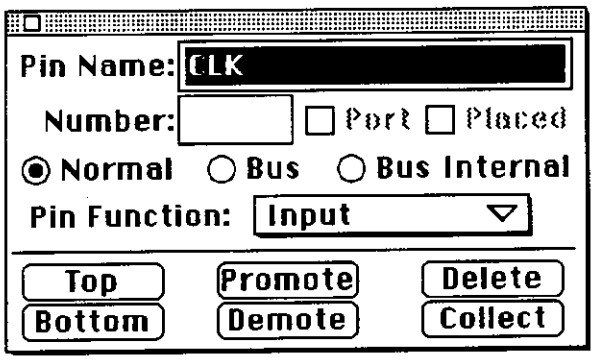
Shift Key Usage

The Shift key on the keyboard serves two purposes in the Device Editor:

- holding the Shift key pressed while clicking on an object adds to the current selection (i.e. selects the object without deselecting the other selected objects).
- holding the Shift key while using the graphical tools may augment the operation of the selected tool.

Pin Information Window

To make the floating Pin Name Info window appear double click on any pin name in the pin name list. The window will display the state of the selected pin names.



The floating palette is divided into two parts. The top is used to examine and modify the characteristics of a pin name. If more then one pin name is selected in the pin name list then the information in the window can not be changed.

The bottom of the pin name info window is used to modify the position and existence of pin names in the pin list. All of the buttons in the lower half of the window affect all of the selected pin names.

Pin Information Fields & Buttons

Pin Name

The name field has room for a 16 character identifier. The name must be unique within its context. The context for normal and bus pin names is all other normal and bus pin names. The context for a internal pin name is all other internal pin names in the same bus, except when the internal pin name is not yet in a bus. In this case its context is the same as normal and bus pin names. This means that regular pin names and/or bus pin names may not be the same, but that internal pin names only need to be unique within their own bus.

If only one pin name is selected in the pin name list then the name field can be changed. The change to the name will be applied when a new selection is made in the pin name list or when a carriage return is typed. If the name

is not acceptable for some reason a beep will occur and the name will not be changed.

If more then one pin name is selected in the pin name list then the name field will show the first pin name selected, but the contents can not be changed.

Pin Number

A four character identifier is displayed in the pin number field. Even though this is referred to as a number it may contain any valid character. For example, "10", "14", "E", and "In" are all valid pin numbers. The value entered for the number is displayed on the stem of a pin.

If more then one pin name is selected in the pin name list then the number field will show the number of the first pin name selected, but the contents of the field can not be changed.

Normal, Bus, Bus Internal Buttons

The three radio buttons marked "Normal", "Bus", and "Bus Internal" show the current type of the pin name. If the pin name is not associated with a graphic pin then the radio buttons are enabled and may be changed. Changing the radio buttons causes the pin name to be displayed differently in the pin name list. If the pin name is associated with a graphic pin then the radio buttons will not be enabled but will display the current state of the pin name.

If more then one pin name is selected in the pin name list then the buttons will show the type of the first pin name selected, but the selected button can not be changed.

Pin Type Pop-up

The pin function pop-up menu allows the type of the pin to be specified. The pop-up will only be displayed for pin names of type "Normal" or "Bus Internal". Pin names of type "Bus" do not have their own type.

Using the pop-up menu the following functions may be assigned to the pin name: Input, Output, Tristate, Bidirectional, Open Collector, Open Emitter, Tied High, Tied Low, Latched Input, Latched Output, Clocked Input, Clocked Output, Clock Input, and No Connect.

The pin type is important to simulation - is this pin driving or driven, and additionally can be important in error checking and report and netlist generation.

See Appendix A - Device Pin Types for more information on pin types.

If more then one pin name is selected in the pin name list then the function pop-up menu will not be displayed.

Port Checkbox

The port checkbox can not be changed by the user and is only used for information. When checked it indicates that the pin name matches a port name in an attached subcircuit. If the box is not checked and you believe it should be then one of the following is the cause.

One, no attached internal or external subcircuit.

Two, no port with the same name in the subcircuit.

Three, the pin name is an internal pin name and the pin name's bus name is not the same as the bus port in the subcircuit.

Placed Checkbox

The placed check box can not be changed by the user and is only used for feedback. When checked it indicates that the pin name is associated with a placed graphical pin.

Pin Name List Buttons

Top & Bottom Buttons

The top and bottom buttons move the selected pin names in the pin name list to the top or bottom of the pin name list, respectfully.

Promote & Demote Buttons

The promote and demote buttons move the selected pin names in the pin name list up or down one pin name in the pin name list, respectfully. If a selected pin name reaches either the top of the list when promoting, or the bottom of the list when demoting, then it will stop and the remaining selected pin names will begin to collect at the top or bottom of the list, respectfully.

Collect Button

The collect button causes all of the selected pin names in the pin name list to be grouped following the first selected pin name.

Delete Button

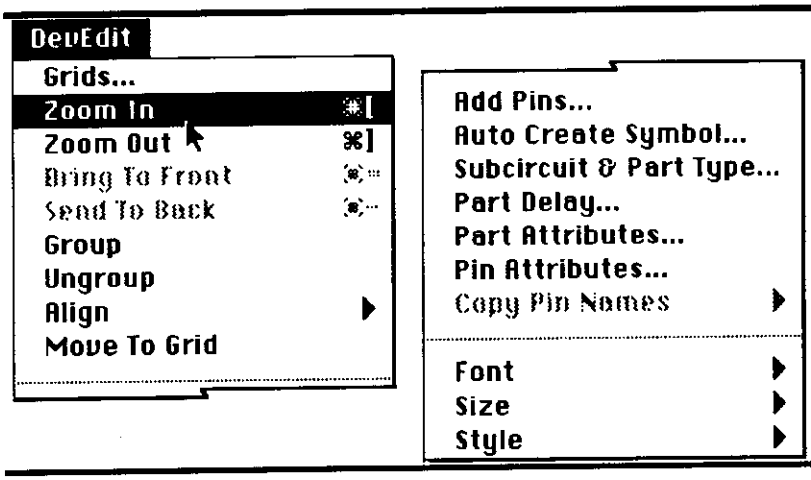
The selected pin names in the pin name list and there associated graphic pins are deleted after the user is prompted to see if this is ok.

Changing the Selection

The pin name displayed can be changed by clicking in the pin name list or a keyboard short cut may be used if the window has the text focus.

Pressing the enter key moves to the next pin name, shift/enter moves backwards one pin name, option/enter moves the selection to the top of the pin name list, option/shift/enter moves the selection to the bottom of the list.

DevEditor Menu Commands

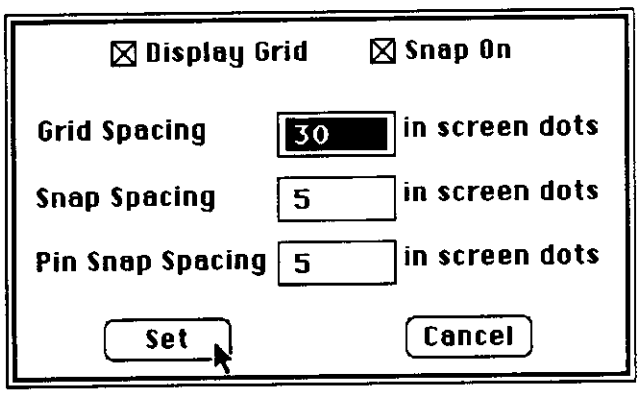


DevEditor Menu

This menu contain all of the DevEditor-specific operations. Each menu item will be discussed below.

Grids

This command allows the user to specify the visible grid spacing and the snap-to grids for objects drawn using the drawing tools.



Display Grid Checkbox	This check box determines whether visible grid lines are shown in the drawing workspace of the DevEditor window. The spacing between these grid lines is determined by the value in the "Grid Spacing" field.
Snap On Checkbox	This check box determines whether the corners of objects made with the drawing tools are moved to the nearest snap-to grid point.
Grid Spacing	This number determines the spacing between the visible grid lines. The units are in screen dots at the default zoom level.
Snap Spacing	This number determines the spacing between snap-to points for the drawing tools (not including pins). This does not affect objects that have already been placed. The units are in screen dots at the default zoom level.
Pin Snap Spacing	This number determines the snap-to grid for device pins. This must be a multiple of 5 to meet the DesignWorks pin grid requirements. The units are in screen dots at the default zoom level.

Zoom In / Zoom Out

These commands allow the user to adjust how close-up an object is viewed. The default setting for the DevEditor is to display objects at the same size as they will appear in the schematic. There are five display levels: 1/2 as large, normal, 2, 4, and 8 times as large.

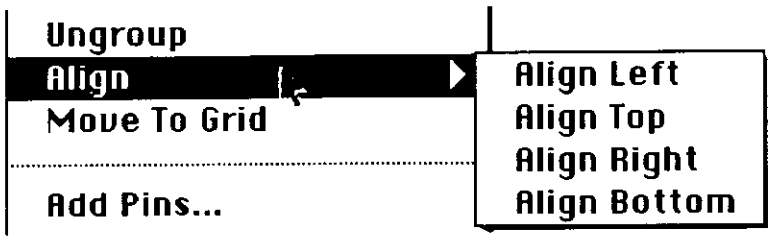
Bring To Front/Send To Back

These commands are used to set the front-to-back ordering of the selected objects relative to the other graphic objects.

Group / Ungroup

These commands allow you to make multiple graphic objects, except pins, be treated as a single object or visa versa.

Align



The Align sub-menu allows you to pick how the selected objects will be aligned. For example, Align Left causes all of the selected objects to be moved such that their left edges are aligned with the left most selected object's left edge.

Add Pins

This command bring up the Add Pins floating palette (See How To Procedures / Adding Pin Names).

Auto Create Symbol

This command brings up the automatic symbol creation dialog (See How To Procedures / Automatically Creating Symbols).

Subcircuit & Part Type

The dialog displayed when this command is selected allows you to specify the type of DesignWorks part being created. The DesignWorks types are: No Subcircuit, Internal / External Subcircuit, Symbol Only, Behavioral, and Primitive (See the How To Procedures section, subsections Selecting the Subcircuit, Assigning a Behavioral Model, and Assigning a Primitive Type).

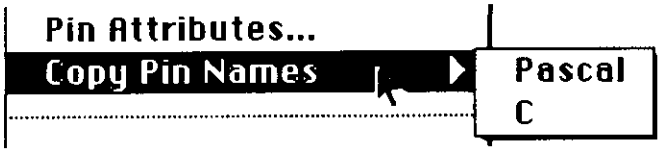
Set Delay

This command allows you the set the default delay for a part. This delay value can always be overridden for each instance of the device in a circuit.

Part and Pin Attributes

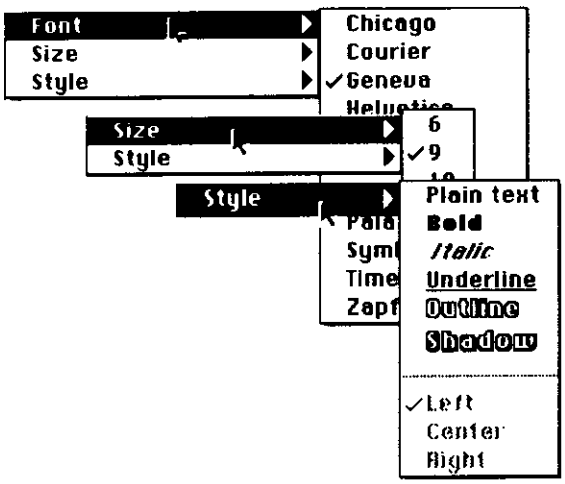
See How To Procedures / Assigning Attributes and Chapter VI - Attributes.

Copy Pin Names



The Copy Pin Names command places a textual list of the device pins on the clipboard. This is used to assist in creating device simulation models written in a standard programming language. Two alternate text formats are available which provide constant definitions for the Pascal or C programming languages.

Font / Size / Style



These menu items set the text characteristics for the selected text objects. If no objects are selected when an item from a menus is selected then the selected text property becomes the new default.

File Menu

New Part

The New Part command opens a new, empty DevEditor window.

Open Part

The Open Part command allows you to edit the part selected in the parts palette. If no part is selected then a dialog will be displayed indicating that a part must be selected first. Parts may also be opened directly from the parts palette (See the Parts Palette manual).

Close Part

The Close Part command closes the current DevEditor window. If any changes have been made to the open part, you will be prompted to save or discard the changes. This is the same as clicking in the go-away box, located in the upper-left corner of the window.

Save Part

The Save Part command saves the contents of the current DevEditor window back to the library it was read from. If the open part was not read from a library (i.e. it was just created), this item will be disabled. (See How To Procedures / Editing an Existing Part / Saving the Part Again).

Save Part As

The Save Part As command brings up a dialog which allows you to select an open library and the name for the part (See How To Procedures / Creating a New Part / Saving the Part).

Edit Menu

The standard Edit menu commands Cut, Copy and Paste can be used to move objects inside and between the DevEditor window, DesignWorks circuit windows, and other applications. Some types of graphic objects created by other programs are not supported by the current version of DevEditor and will not appear if pasted into the DevEditor workspace area.

DevEditor Procedures

Creating a New Part

NOTE: This procedure describes the creation of the simplest possible device symbol. If you are using the gate packaging or auto-naming features of DesignWorks, refer to the section Adding Packaging Information, below.

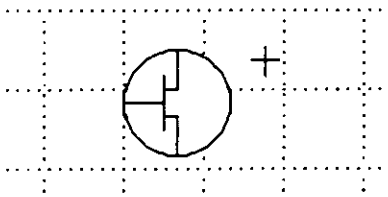
To create a new part, select the DevEditor item in the Tools menu. If there were no open DevEditor windows then this will open an empty drawing window with a default name and an empty pin list.

If there already existed a DevEditor window then selecting DevEditor in the Tools menu brought this window to the front. You can get a new DevEditor window by selecting "New Part" from the File menu or by closing all open DevEditor windows and again selecting DevEditor from the Tools menu.

Drawing the Graphics

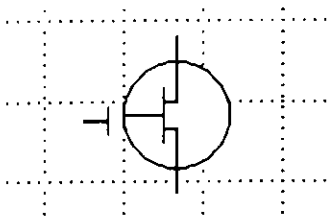
Draw the graphical shape which represents the part. Do this using the line, rectangle, rounded-rectangle, oval/circle, arc, and polygon tools.

NOTE: Device pins must be added using the pin tools. Do not draw any of the part's pins with the graphic tools mentioned above.



Adding Pins

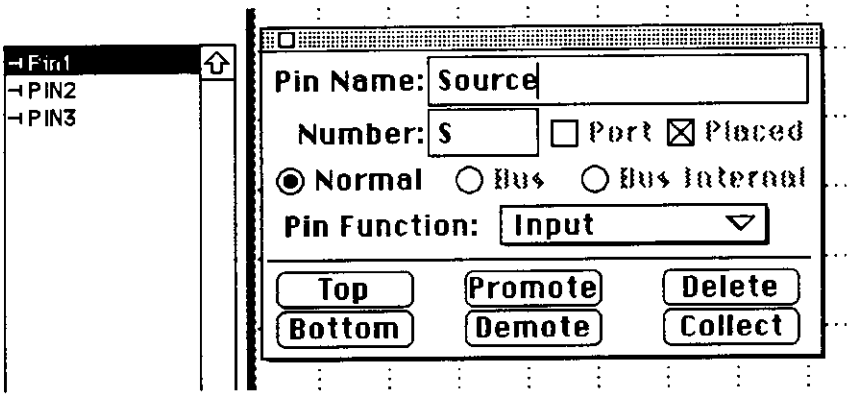
Place pins on the part use the 12 pin placement tools. The first row allows placement of normal pins. The second row provides pins with inversion bubbles. The last row of pin tools are used to place bus pins.



Note that each time you place a pin a new default pin name is added to the pin name list (See Part Pin Tools for more details).

The pin name and other pin information can be changed using the pin name info window (See Pin Name Info Window). For this example we want to name the pins "Source", "Gate", and "Drain". In addition we want to give them the following pin number "S", "G", and "D".

To change the information about each pin we first bring up the Pin Name Info Window by double clicking on a pin name in the pin name list. Double click on the first pin. When placing the pins we placed the top pin, then the bottom pin and finally the pin on the left. Therefore, the names in the pin name list are in the same order.



Enter the name "Source" and the number "S" for the first pin we placed. You can use the tab key to move between the name and the number field. When you have entered both the name and number - Press the Return key. Notice that the name was updated in the pin name list and that the pin number "S" has been displayed with the graphic pin.

To change the next pin you can; click on the graphic pin; click on the pin name in the pin name list; or if the Pin Name Info Window still has the text focus press the Enter key. Press the Enter Key. Pressing the enter key also causes the pin information to be copied to the pin, just like pressing the return key.

Notice that the pin whose information is displayed in the Pin Name Info window is also selected in the pin name list and if it has been placed then its graphic image is also selected. This allows easy verification that the correct pin is being renamed.

You should now see PIN2 selected in the pin name list, the bottom graphic pin selected, and the name PIN2 in the Pin Name Info window. Do as describe above for each of the remaining pins.

When you are finished with the Pin Name Info window it may be simply closed by clicking on its go-away box or since it is a floating window it can be just moved to the side.

Note: The actual names given to the pins in this simple form of part are not critical. It is a good idea though to use meaningful names for the following reasons:

- These names will be seen in the Schematic tool's pin info dialog.
- They can be extracted in netlist output.
- They are used when binding pins to ports in subcircuits.
- In the case of bus internal pins, they are used when connecting bus pins on the part to busses.

Creating a Part With Subcircuit

Port Interface

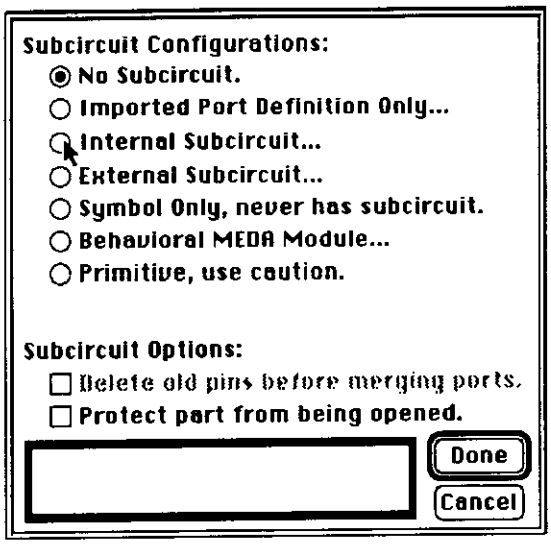
A part's subcircuit is a schematic circuit which is associated with the part such that it is considered to be the part contents. A circuit which is to be used as a subcircuit must include parts called port connectors. Port connectors, which are named, allow signals in the circuit to be associated with pins on an enclosing part. Port connectors make this association by name, i.e.: a port connector named "A0" will only associate with a pin with the same name.

See Chapter VII - Hierarchical Design for more information on the port interface.

Selecting the Subcircuit

The easiest method to create a part with an associated subcircuit is to begin by selecting the schematic circuit to be used as the subcircuit. This sounds like stating the obvious, but creating the part in any other order is more work.

To select the circuit to be used as the part's subcircuit you must also define the type of part being made. Select the Subcircuit & Part Type command from the DevEditor menu.



This dialog allows you to select among several different options:

No Subcircuit	This is the default. It indicates that the part being made has no associated subcircuit, but it doesn't rule out that a subcircuit may be attached by the schematic capture section of DesignWorks.
Imported Port Definition Only	This is similar to the No Subcircuit option. No subcircuit becomes associated with the part, but a port interface is extracted from a circuit for use with the part. Selecting this option prompts you to pick an open schematic circuit. The port names are read from the circuit without attaching the circuit to the part. Like the above option, this doesn't stop a subcircuit from being attached later. Since a subcircuit wasn't attached to the part, the port interface is only associated with the part for the length of the editing session.

Internal/External Subcircuit	These two options prompt for an open circuit and a saved circuit, respectfully. The selected circuit becomes associated with the part being created as its subcircuit. The difference is that the Internal Subcircuit option includes the circuit definition as a piece of the part and saves that piece with the part in the library. The External Subcircuit option only includes a file reference to the circuit file with the part in the library. Both options import the circuit's port interface so that the names of the ports are available in the DevEditor.
Symbol Only	This option is like the No Subcircuit option except its doesn't allow the schematic capture part of DesignWorks to associate a subcircuit with the part in the future.
Behavioral & Primitive	(See their related section of this manual).

There are two subcircuit options in the lower part of the dialog:

Delete old pins before merging ports	This option is only enabled when the selection made above results in a port interface being read. If this box is checked when the "Done" button is pressed then all of the old pin names in the pin name list will be deleted. This allows a new port interface to be brought in without any conflicts with the existing pin name list. If this option wasn't checked then port names would be merged with the names in the list. Duplicate pin names and their related properties remain unchanged, except they are now associated with the new port. Unmatched pin names in the pin name list remain exactly the same.

Protect part from being opened	This option has the effect of saying, "Yes. There is a subcircuit, but in general you don't want to go into it". This causes the schematic capture part of DesignWorks to prompt to make sure it is really OK to enter the subcircuit before doing so. It also controls if the report generator will list the contents of this device in a netlist. In general this is used for parts like 7400 series devices. Where a designer treats the part as a physical component, but there may be a subcircuit for the simulation purposes.

If option 2, 3, or 4 was selected in the dialog then upon exiting from the dialog a port interface will be read and the pin name list in the DevEditor window will be updated to show the new pin names. In addition, if option 3 or 4 was selected then a subcircuit is also attached to the part being created in the DevEditor window.

Drawing the Graphics

The graphic image of the part may be drawn in the same way as was described in the previous section on creating part's without subcircuits.

You can also generate a symbol automatically using the Auto Create Symbol command. See the section Auto Creating a Symbol.

Associating Graphic Pins with Pin Names

Pin names in the pin name list that may be placed are those that are not indented and that don't have a symbolic pin symbol next to their names. Names that have a symbolic pin next to them already have a graphical association. Names which are indented are internal bus pin names and don't have a graphical representation.

Assignment of graphical positions to pin names may be done explicitly by using the pin placement tools or automatically using the Auto Create Symbol command. See Auto Creating a Symbol, below.

To explicitly assign a graphical pin position to a pin name requires that the pin name be selected in the list and that a graphical pin be placed using one of the pin tools.

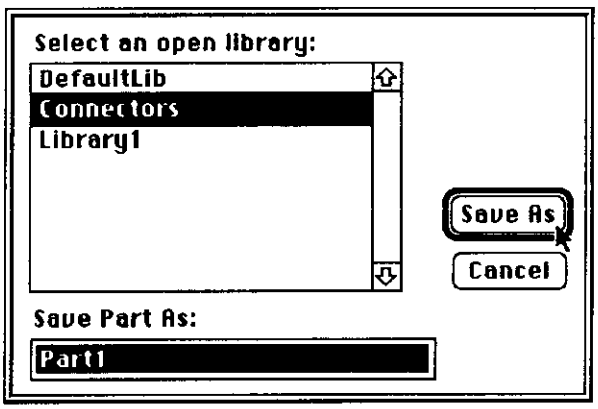
A graphical pin assignment will only be made with an unplaced pins of the correct type. If a placed pin name is selected in the pin name list, and an

attempt is made to associate a graphical pin with it, then the association will not be made with the pin name because it was already placed. Instead the new graphical pin will be associated with the next free pin name of the correct type or a new pin name will be created if there are no unplaced pin names of the correct type.

Because of the way the pin association mechanism works, i.e.: searching for the next placeable pin name, it is possible to place all pin names of the same type without having to make a new selection. Select the first unplaced pin name and place it. Now since the current selection is placed, the next graphic pin placed will be matched with the next pin name of the correct type. Place another pin and note how the selection moves forward to the next previously unplaced pin name of the same type.

Saving the Part

The newly-created part is written out to a library using the Save Part As command in the File menu.



When the menu item is selected the "Save Part As" dialog will appear. It requires that a library name is selected from the list and that a name for the part is entered in the lower box. In this example the connectors library has been selected and the part's name has been left to default to "Part1". A name entered will become the name of the part in the library and the title of the DevEditor window will be updated to correspond to the new part name. Note: only open libraries appear in the list (See Library Handling in the schematic capture manual for additional information).

Editing an Existing Part

Getting the Part from a Library

The following methods can be used to edit an existing part in a library:

- a part selected in the part palette may be edited by pressing the mouse button with the ⌘ key down and then selecting "Edit Part" from the pop-up menu. The mouse must be over the parts list for the pop-up menu to appear. Another place to find this menu is under the File/Libraries menu pull-right.
- open the Device Editor using the DevEditor item in the Tools menu, then use the Open Part command in the File menu.

The part once opened is a copy of the part in the library. The part in the DevEditor window knows what library it can from and the name of the part, but it is not the same part as in the library. If a part is selected from the library while the same part is being edited the original part will be the one selected. In addition, editing a part and returning to the library does not change all the copies of that part in all the circuits that use it (See the schematic manual on methods on updating from a library).

Warnings About Editing

Changing Graphics

Editing changes can be freely made to all of the part's graphical components other then pins. It is important to remember that graphic pins are associated with pin names and that even though graphic pins look the same they may not be interchangeable because of their associated name.

Consider the following example. A flip flop has pins named "D", "CLK", and "Q". The pins "D" and "CLK" are both left facing graphic pins and "Q" faces to the right. Simulation and netlisting problems will occur if the graphic pins "D" and "CLK" were to have their graphical locations swapped.

Swapping the pins' positions would confuse the user because the "D" label would be located next to the graphical pin associated with the pin name "CLK".. The same would be true for the "CLK" pin.

To ensure that pins are placed correctly it is important to select the pin in the pin name list and verify that the expected graphic pin is selected.

Changing Pin Information

If the part has a subcircuit then it is important to remember that the pin types are derived from the ports in the subcircuit. Trying to change a pin's type without also changing the underlying subcircuit is not allowed.

For parts based on a primitive types or behavioral models the functions of the pins must agree with the selected model. There is currently no check to ensure that the pin are set correctly for the model selected.

With other types of parts, Symbol Only and No Subcircuit, the types of the pins can be changed freely.

Changing Part Type

Changing a part's type can have dramatic effects. It is recommended that a part who's type has been changed is never saved into the same library with its original name. Doing this ensures that mistakes are easier to recover from. Some of the things to look out for are:

- Changing an internal subcircuit to any other type. An internal circuit is kept with the part in the library, it is important to remember that changing the part's type will cause the circuit to be deleted. This is a problem if the circuit was going to be used again and there is no copy of the circuit.
- Changing to an internal or external subcircuit. When a circuit is associated with the part its ports are read and used to create pins with the same name. These new pins carry the type of the port and will replace like named pins that are already defined. This will result in a pin losing the type it was initially defined with.

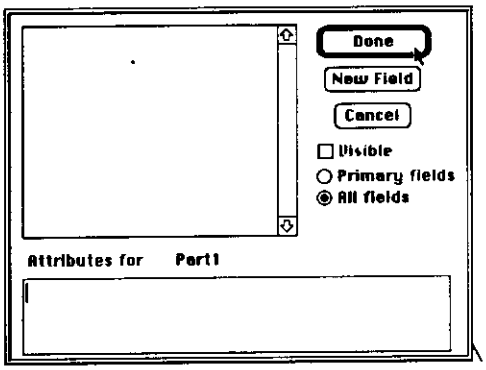
Saving the Part Again

A part opened from a library knows what library it came from. The Save Part command in the "File" menu replaces the original part in the library with the one being edited. This operation can not be undone so it is not wise to replace the original part unless you are sure it is correct. If you are not sure the part is correct, then it is better to save the part to a different library or with a different name to the same library. This allows you to test the part completely without loosing the original.

Assigning Attributes

Editing Part & Pin Attributes

To edit the attributes associated with a part select the Part Attributes command in the DevEditor menu. To edit the attributes associated with a part's pins select a pin in the pin name list and then select the Pin Attributes command from the DevEditor menu. The only visible difference between the part attribute dialog and the pin attribute dialog is the addition of "Next" and "Previous" buttons at the bottom of the window.



Attributes defined while in the DevEditor will be associated with the part in the library. These will become the default attributes for the part when it is placed in a circuit.

For more information on how to create and modify attributes see the appropriate section in the schematic manual.

NOTE: There are numerous attributes which have special meanings and conventions for usage. Refer to Chapter VI - Attributes for a complete discussion of the setting, modification, and interaction of attributes.

Adding Pins

The Add Pins command in the DevEditor menu causes a floating palette to be displayed. Pin names the user wishes added to the pin name list, and optionally their pin number, may be entered into the palette. The pin names are created and merged with the contents of the pin name list when a carriage return or enter key is pressed or when the Add button is clicked. The Add Pins palette does not create graphic pins only pin names for the pin name list.

The created pin names are merged with the names in the pin name list. If a like named pin already exists in the list then it may be reordered to appear in the same order as in the Add Pins palette. If a pin being added has a pin number defined then this pin number will replace the pin number in the like named pin.

Syntax for Pin Names

Pin names may be up to 16 characters long. There are three types of pin names which may be described. They are:

- **Normal Pin Names** may be specified as individual names, e.g.: A B C D, or as sequences, e.g.: A0..3. After each pin name or pin name sequence an optional pin number specification is allowed.

  A pin number specification defines the pin number(s) associated with the proceeding pin name or pin name sequence. The specification starts with an (and ends with an). For a single pin name the pin number specification should contain a single number. For a pin name sequence multiple numbers and sequences may appear between the brackets.

  In the picture above CLK and OE both define normal pin names which have pin numbers 10 and 17, respectfully. The sequence A0..7 defines the pin names A0 through A7. These pin names were also given pin numbers. The pin numbers 1 2 3 4 11 12 13 and 14 were assigned.

  It is also possible to skip a pin name when assigning pin numbers to a pin name sequence. Consider the previous example, if we didn't want to assign the number 11 but wanted all other pin names to get the same number, we would do the following: A0..7(1..4,,12..14) instead of A0..7(1..4 11..14). Two commas in a row causes a pin to be skipped when assigning the pin numbers. Three commas cause two pins to be skipped. Four commas, etc.

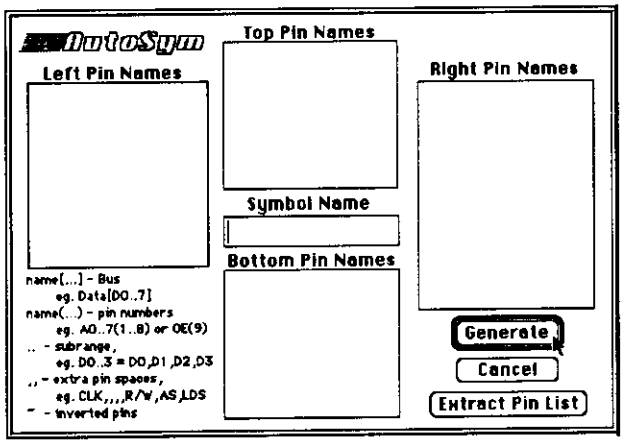
- **Bus Pin Names** are denoted by a name followed by [...]. Bus pins do not need to have internal pin names defined and may not have pin numbers.

- **Internal Pin Names** must appear between a [and a]. They have the same format as normal pin names and may have pin numbers. Internal pin names may be added without adding a bus name pin by not placing a bus name in front of the []. For example, [A B C] adds the internal pin names "A", "B", and "C" to the pin name list.

Automatically Creating Symbols

The Auto Create Symbol command invokes the Automatic Symbol Generator which will create standard rectangular symbols given a list of the desired input and output pin names. For maximum flexibility, the symbol generated consists of separate graphic objects and is completely editable after it is generated.

The current settings for line width, fill patterns, color, text font, size and style, etc. are used in generating the symbol. The only exception to this is the type name text placed in the center of the symbol, which is written 3 points larger than the current setting and in bold.



Selecting the Auto Create Symbol command displays the above dialog box. The pin name boxes will contain the information entered the last time the DevEditor was invoked for this part. These can be modified as desired. The new settings will be merged with pin name list when the Generate button is pressed.

Entering Pin Names

The four pin name boxes allow you to specify the names of pins to appear on the left, right, top and bottom of the device symbol. The syntax described in the previous section, Adding Pin Names, is also used to define the pin names and numbers with the following extensions:

- inverted pins can be specified using the "~" character in front of the name. The inverted pin will be displayed with a bubble. The "~" will not appear in the symbol.

• items in a list can be separated by blanks or commas. Placing an extra commas between two items adds extra space between the pins on the symbol. Additional space can be added with more commas.

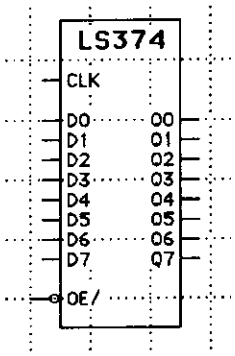
Entering the Symbol Name

The symbol name text box allows you to specify the text that will appear centered at the top of the symbol. This also becomes the new name for the part.

Generate Button

The Generate button causes the current contents of the active drawing window to be erased and replaced by the generated symbol. The pin name list will be merged with the new pins described in the dialog. This symbol consists of standard graphic objects so it can be edited using any of the drawing tools provided.

An example of a device produced by the Auto Create Symbol:

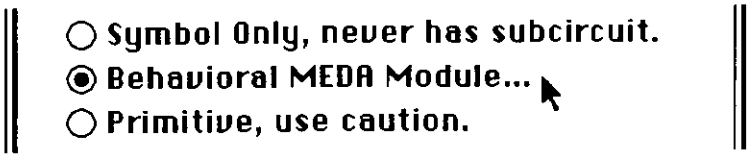


Extract Pin List Button

The Extract Pin List button updates the pin name boxes with the names extracted from the pin name list. Pins which are inputs and busses are placed on the left, outputs are placed on the right.

Assigning a Behavioral Model

A Behavioral Model (i.e. a part simulation model written in a programming language using the MEDA interface) can be assigned to a symbol in a manner similar to an internal circuit.



Choose the Subcircuit & Part Type command in the DevEditor menu, then select the "Behavioral MEDA Module" radio button. A standard open file box will then appear allowing you to select the file containing the compiled behavioral model code. This code will then be loaded from the selected file and included in the library with the part.

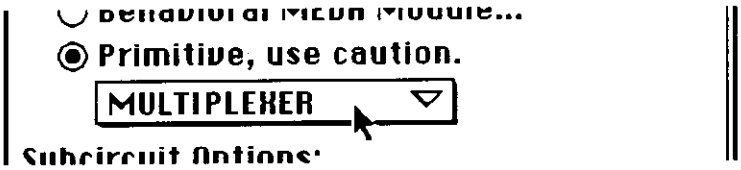
It is the user's responsibility to add, place and configure the type of all the pins assumed by the behavioral model code. The "Copy Pin Names" command can be used to create a list of the pin names in the symbol for use in creating the model code.

Creation of simulation models requires the MEDA Developer's Kit and the Macintosh Programmer's Workbench.

Assigning a Primitive Type

NOTE: The Primitive type settings should only be used with a clear understanding of their functions. Primitive types are intended primarily for use with the DesignWorks Digital Simulation option. See Appendix B - Primitive Device Types.

The primitive type of a part determines which of 38 different internal simulation model will be used to simulate the part. To describe parts which are larger or more complex a subcircuit should be used (See How To Procedures / Creating a New Part / With A Subcircuit).



To select the primitive part type choose the "Subcircuit & Part Type" item in the DevEditor menu. Then select the "Primitive" radio button. This will cause a pop-up menu to be displayed below it. The pop-up menu will show the primitive type currently selected. Clicking on the pop-up menu will display the other options available (See the Appendix / Table 1).

Creating a Gate

NOTES:

1) This section applies only when creating a gate for use with the DesignWorks Digital Simulator option. For general schematic use, use the default primitive type setting.

2) See information on entering gate packaging information later in this chapter.

To make an inverter (NOT gate) place 2N pins. The order must be input A, output A, input B, output B, ... Then set the primitive type to NOT. This will create a part with N inverters.

To make an AND, OR, XOR, or XNOR gate place N+1 pins. The first N pins must be inputs and the last pin an output. Then set the primitive type to the correct logical gate. This creates a single logical gate performing the logical function selected.

Creating a Breakout

A breakout is a special device that allows signals to be associated with a bus. It consists of 1 Bus pin, no internal pins, and N normal pins. Because the device type is breakout the normal pins will be connected to like named signals in the bus.

Note: Another way to split signals out of a bus is to use a part with a subcircuit as a splitter, which explictly routes signals between pins. This has advantage of flexibility since signals do not have to be explictly broken out but may instead be split into busses or any combination of busses and signals. The disadvantage is that since the splitter is a device it will be listed in hierarchical netlists. In flattened netlists it will not appear if marked as a non-protected device. In addition, there is a significant memory and file size penalty.

Creating a Power (Signal) Connector

A power or signal connector is a special type of device which is generally used to represent a power or ground source, e.g.: +5V, +15V, -15V, -5V, Ground, Vss, Vdd.

These devices have a special properties in the schematic. When a pin on one of these devices is connected to a signal it attempts to assign its pin

name to the signal. If the signal doesn't have a name then it gets the name of the pin. If the signal is named, and the name is different from the pin's name, then the user will be prompted to select between the signal name and the pin name.

An additional property of signal connectors is that any signal they are connected to is exported across all pages of the schematic.

The type of the pin must be set correctly if simulations are to make sense. For signal sources like +5V, +15V, and Vss, a normal simulation would expect a logical value of 1 (True). For signal sources like Ground and Vdd, a logical value of 0 (False) is expected. This means the pin type should be "Tied High" or "Tied Low" (See Pin Name Information Window / Pin Information Fields & Buttons / Pin Function Pop-up).

If you don't want the signal connector to supply a signal value, but only its name and the fact that it makes the signal global, then the pin type should be set to "Input".

Creating a Port Connector

Port connectors have the property of associating a signal or bus with a pin on an enclosing part. The association is made by name. The port's Name is compared to the pin names on the parent part. Internal pins in busses are matched by pin name.

See Chapter VII - Hierarchical Design for more information on port connectors.

For a Signal

A port connector for a signal must have a pin which is of the correct type to interface the signal to the parent part's pin. For example, an input pin on the port connector would be correct to connect to a output pin on the parent part. The name of the port connector pin is not important. Only the name assigned to the port connector once it is placed is important; it must be the same as the parents pin.

For a Bus

A port connector for a bus must have a bus pin which contains pins of the correct type. For example, a bus pin with three internal pins A (input), B (input), and C (output) would be correct to connect to a parent part's bus pin which contained pins A (output), B (output), C (input). The name of the

port connector bus pin is not important, but the internal pins must have the same name. Once the port connector is placed in a circuit its reference name is important, it must be the same as the parent's pin name.

Creating a Page Connector

Page connectors have the property of exporting the signal or not there connected to across all pages.

For a Signal

A page connector for a signal contains only input pins. The name of the pins is not important.

For a Bus

A page connector for a bus contains only bus pins with no internal pins. The name of the bus pin is not important.

Adding Packaging Information

A number of DesignWorks features rely on certain attribute fields being present in a part definition. If you create a part symbol with no default attributes it will have the following characteristics:

- The Packager will assume that it has only one unit per package, i.e. a unique package name will be assigned to each device and the Unit field will be left empty.
- If you have not assigned default pin numbers to the pins, they will have to be entered manually for each device on the diagram.
- Part number, component value and package attributes, if needed (e.g. for PCB netlists), will have to be entered manually for each device on the schematic.
- No power and ground connections will be made automatically.
- The default prefix will be used to auto-generate device name when the part is placed.

The following sections describe the attribute entries that should be made in each part created in order to take advantage of these DesignWorks features.

Gate Packaging

For any part that can have multiple units assigned to a single package, the following fields should be set up:

Field Name	Set In	Description
Unit	Device	This field gives the default Unit setting when a device is placed and can be omitted, if desired. In any case, if Auto Packaging is enabled, it will be overridden by the Unit setting assigned by the Packager.
Unit.List	Device	This field contains a list of the units that can be represented by this symbol. The units are normally specified by a single lower-case letter, although any names up to 16 characters can be used. Items should be separated by commas, e.g. "a,b,c,d".
Unit.All	Device	This field contains a list of all units that can be present in this part type, i.e. including those represented by different symbols. This can be omitted if it is the same as "Unit.List". It will only be different for parts that have multiple functions that can be in the same package.
VisPin.List	Each pin	This field contains a list of the pin numbers that can apply to this pin. There must be the same number of items in this list and in the same order as the Unit.List field in the device. E.g. "1,4,10,13".

See a more detailed description of the packaging mechanism and fields in Chapter V - Advanced Schematic Editing.

Auto-Naming

Both the Packager and the Auto-Naming features make use of attribute fields in the generation and positioning of names for devices.

NOTE: DesignWorks allows you to select any device field to be used as the prefix for default name generation. The prefix fields described here are the standard ones included in the DesignWorks libraries, but you can also define your own if you have special naming requirements. See the description of the PrefixField design attribute elsewhere in this manual.

Field Name	Set In	Description
Name.Prefix	Device	This field gives the prefix to be used in generating a device name or package name. In the DesignWorks standard libraries, this field is empty for most integrated circuit parts. They therefore take the default prefix specified with the design. For discretes, the common prefix character is specified.
Name.Spice	Device	This field is specified only for symbols to be used with a SPICE-based analog simulator. It is an optional name prefix matching the standard required for SPICE models.
Name.Pt	Device	This field contains an X,Y coordinate for the default position of the name when the device is placed. This is specified in 1/1000 inch and is relative to the top left corner of the symbol. E.g. "1000, -200". See Chapter V - Advanced Schematic Editing for more options in this field.

Part and Package Fields

The following fields are the standard ones used to specify the part number and package type for netlisting purposes.

NOTE: The standard part and package fields are listed here. You are free to add new fields for special purposes at any time and modify the field names used in the netlist formats.

Field Name	Set In	Description
Part	Device	This field gives the part name that will be used for netlisting purposes. This may be the same as the name as the part is saved under in the library, or it may be in a format required by some external package.
Part.List	Device	This field contains an optional list of part numbers that can be represented by this symbol. If specified, this list will appear in a value list menu in the device pop-up menu on the schematic.

Part.Pt	Device	This field contains an X,Y coordinate for the default position of the Part field when the device is placed. This is specified in 1/1000 inch and is relative to the top left corner of the symbol. E.g. "1000, -200". See Chapter V - Advanced Schematic Editing for more options in this field.
Package	Device	This field specifies the default package type for use by an external PCB layout package.
Package.List	Device	This field contains an optional list of package codes for the device. IMPORTANT: This field is unusual in that it is linked in DesignWorks to the Part.List field. If an item in Part.List is selected by the user, the corresponding item in Package.List is automatically placed in the Package field.

Other Fields

The following fields are included in the standard DesignWorks libraries although they are not connected with any specific program features.

Field Name	**Set In**	**Description**
Power	Device	This field contains a list of pins to be connected to the power net in netlist reports.
Ground	Device	This field contains a list of pins to be connected to the ground net in netlist reports.
Value	Device	This field contains the default component value for the part. It normally only applies to discrete components.
UnusedPins	Device	This field contains a list of the unused pins in this part type. This field is not included in any of the standard DesignWorks report formats, but is included for general usage.

This page is intentionally left blank.

Chapter XI - Report Generation

Introduction to Report

The DesignWorks Report tool works with the DesignWorks schematic entry tool to provide flexible text report generation capabilities.

Report Features

- Operates from within DesignWorks on any currently open design file.
- Produces the following types of reports, with numerous sorting and format options:
  - Netlists by signal or device
  - Bills of materials
  - Signal and device lists with graphical data
- Produces text files that can be read by any text editor, word processor, or spreadsheet or transferred to external design packages on the Macintosh or other machines.
- The detailed format of the output file is determined by a "form" file, allowing a wide range of output formats to be implemented by the user.
- Works with the DesignWorks "attributes" feature to allow arbitrary text data to be stored with any object and placed in a specified position in an output listing. This can be used for data such as special part numbers, manufacturing data, simulation parameters, etc.
- All types of listings can be sorted and provide counts based on name, attribute field value or other object parameters.
- Includes a number of standard form files for commonly-used formats. These can be used to interface with other design packages, or as samples for developing your own form files.

General Information on Report

The Report tool produces circuit information text files in a wide variety of formats. Essentially any information about a circuit can be placed in an output file.

Output File Formats

Report creates text files which are intended to be incorporated into other documentation or transmitted to other systems. Files are saved with a file type of "TEXT", so they are accessible to almost any word processing program or text editor that runs on the Macintosh.

If it is desired to print any of these files, you will generally get the best results by converting the file to a font such as Monaco or Courier, which has fixed character spacing. No font information is stored with the files.

Output File Name

The default output file name is the design file name with " Report" added. However, the form file can set any desired file extension and the user can enter any name before the file is saved.

IMPORTANT NOTE: Throughout this chapter we refer to names of design software systems produced by third parties. Many of these names are trademarks and are the property of their respective holders. We can take no responsibility for the accuracy of information in this manual relating to third party systems, as their specifications are subject to change without notice.

Installation

The following two files must be placed in the same folder as the DesignWorks program for correct operation of the report generator:

- the file "Report" contains the actual report program code. It is located automatically when DesignWorks is first started. The Report file cannot be executed independently from DesignWorks.
- the file "DesignWorks Report Form" specifies the format of the output files generated by Report. If this file is not present in the folder, no default form file will be selected when you select the Report command. You can select any other file by clicking on the Select Form File button in the Report dialog box.

The format of the form file is discussed in detail later in this manual.

Using Standard Report Formats

A number of standard report forms are included with the package for use with popular PCB layout, simulation and other design packages. To use one of these formats, click on the Select Form File button after selecting the Report command in the Tools menu. You will then be prompted for the name of a report form file. If you use the same form file consistently, then rename it to "DesignWorks Report Form" and place it in the same folder with the DesignWorks program and it will be selected as the default format when Report is invoked.

IMPORTANT: Many of the standard formats provided with DesignWorks rely on specific attribute data being entered in the design. Use the Format Notes button described below to get specific details on each form file.

Viewing Format Notes

Any of the standard form files supplied will contain a number of comments explaining the format commands used. Some form files may rely on certain attribute fields being present in devices in the circuit. These notes can be viewed using the Format Notes button in the Report box:

A technical description of some of the standard formats is provided in the Examples section later in this chapter.

Form File Naming Convention

All form file names should contain the word "Form" so that they will appear in the Select Form File list. In addition, the following keywords are used to indicate the hierarchical format:

Name Keyword	Example	Description
Flat	Douglas Flat Form	For use with Flat mode designs or to list the current circuit only of a hierarchical design.
Flatten	SPICE Flatten Form	For use only with Physical mode designs. Produces a flattened netlist.
Hier	SPICE Hier Form	For use with Physical or Pure mode designs. Produces a pure hierarchy netlist, i.e. one with separate definitions for nested sub-circuits.

Common Changes to Standard Report Forms

It may be desirable to make minor modifications to one of the standard report forms in order to tailor it to your design needs. This section covers some common changes that can be made without an in-depth understanding of the form file format. Each of these features is covered in more depth in the following sections of this chapter.

Form files can be viewed and edited using a simple utility such as TeachText or any standard word processor. If using a word processor, be sure to save the file using a "Text Only" option so that no extra formatting information is included.

Power and Ground Attribute Connections

The Report tool allows you to create common connections in a circuit using attribute entries in devices. This can be used to create power and ground entries in the netlist without having to show all these connections explicitly on the diagram. The standard form files normally contain entries specifying the Power and Ground attribute fields as sources for power and ground pin lists.

Any device attribute field may be specified as a signal source by the form file command:

```
$SIGSOURCE(signalName) &fieldName
```

- "signalName" is the name of the signal to attach pins to in the netlist.
- "fieldName" is the name of the attribute field to search for in each device to look for pin numbers to attach to the named signal. If &fieldName is omitted, the signal name is taken to be the name of the attribute field to search.

The named device attribute field is assumed to contain a list of pin numbers to attach to the signal. This can consist of a single pin entry, such as "7", or a list, such as "5,6,9,14". For example, most standard report form files contain the following entries:

```
$SIGSOURCE(+5V) &Power
$SIGSOURCE(Ground)
```

There is no fixed limit to the number of SIGSOURCE entries that can be created.

Default File Name

The default output file name is determined by the $CREATEREPORT command in a form file. Following is a typical example:

```
$CREATEREPORT($DESIGNNAME.NET) $PROMPT
```

The contents of the parentheses determines the default name of the output file. The keyword $DESIGNNAME substitutes the name of the current design file. All other characters inside the parentheses are used verbatim. The $PROMPT keyword indicates that a standard file box should be displayed to confirm the file name. If this keyword is not present, the file is saved in the same folder as the design without any prompt.

Attribute Field Usage

Attribute field data is heavily used in generating netlist output files. Most of the standard form files are set up to use the pre-defined attribute fields. These are easily modified to use customized fields which you have added to your design.

Attribute fields are referenced in a form file using the form:

```
&fieldName
```

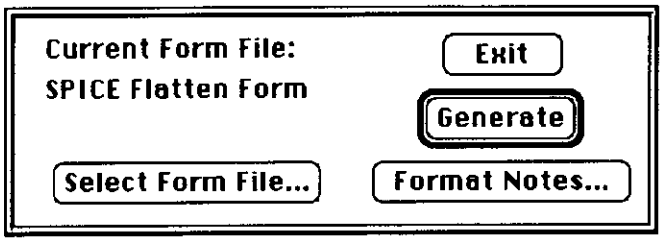
Wherever this format is used, any other field name can be substituted. For example, a typical device listing will appear as follows in a form file:

```
$DEVICES$DEVNAME &Part
```

The "&Part" item will be substituted with the contents of the Part attribute field. Any other field name can be referenced as desired, or additional ones can be added to the line.

Report Operation

The report generator is invoked by selecting the Report item in the Tools menu. The following box will appear:



If the report generator cannot locate the file "DesignWorks Report Form" then there will be no default form file selected and the "Generate" button will be disabled.

Selecting a Form File

The Select Form File button allows you to choose the format of the report. Clicking this button will bring up a standard file box displaying all text files with names ending in "Form". Select the desired form file by double-clicking on it.

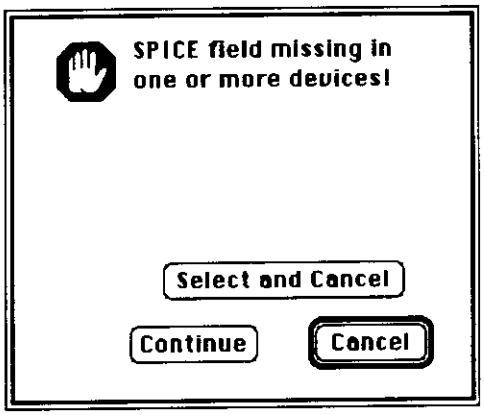
NOTE: Normally, the Select Form File option only shows files with names containing the word "Form" to distinguish them from output files. You can select any text file, regardless of name, by holding the Option key on the keyboard while clicking on the "Select Form File" button.

Generating a Report

Once the form file is selected, the "Generate" button will be enabled and you can proceed with report creation by clicking this button. The procedure after that is determined by the form file. Most form files will put up a standard file creation box to allow you to specify the name of the output file.

Circuit Checking

The report form file can contain optional commands to check for errors or missing data in the circuit before proceeding with generating the report. If such a check is encountered and a circuit error is detected, the following type of box will appear, although the text of the error message depends entirely on the form file:



This box gives you the option of continuing despite the error, canceling the report, or selecting the offending devices or signals on the circuit diagram, then canceling. Note that the "Select and Cancel" option will only select objects on the current (topmost) circuit page, if there are multiple pages in the circuit.

Device Reporting Options

The listing of devices and device internal circuits can be controlled using the Report Options button in the Get Info command for a device. The following options are available:

Omit from report	Nothing about this device or hierarchical block will appear in the netlist.
Report this device	This device or block will appear in the netlist as if it has no internal circuit.
Report sub-circuit	The internal circuit of this device will appear in the netlist. I.e. in a Pure Hierarchical Netlist, the internal circuit will be defined, in a flattened netlist, the internal circuit will be substituted and this device will not appear at all.

See more information on hierarchical netlists later in this chapter.

Report Errors

Two general types of errors can occur in generating a report:

- form file syntax errors
- circuit data errors

In general, no error messages are issued for incorrect syntax in a report form. Any items in the form file that are not recognized are assumed to be literal text and are transmitted to the output file unmodified.

A number of circuit data errors can occur during report generation, including unnumbered pins or excessive string length on data items such as pin numbers. If any of these errors occurs, a warning box will appear with an explanation of the error.

Report Form Examples

This chapter provides some detailed examples of report forms. To illustrate the use of various options, most of the following examples will refer to the following sample circuit:

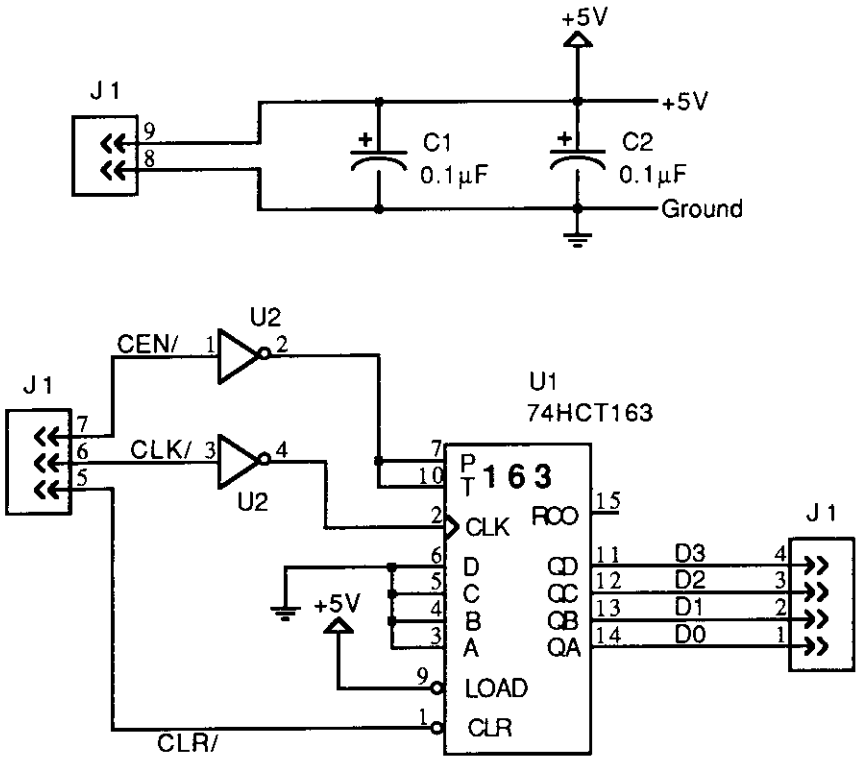


Fig. 1 - Sample Circuit for Report Generation

Note the following characteristics of this circuit:

- several devices have the same name and are intended to be considered as a single device in the reports.
- the connectors labeled "J1" each have a Part attribute entry "Molex 9-pin".
- the inverters labeled "U2" each have a Part attribute entry "74HCT04".
- the capacitors C1 and C2 each have Part attribute entries "Tantalum" and Value attributes "0.1µF".
- the devices U1 and U2 have Power and Ground attributes specifying the standard power and ground pins.

Basic Netlist Formats

A netlist (i.e. a list of the signal connections in a circuit) can take one of two general forms:

- a listing of the signals (or nets) making up the circuit with each signal name followed by a list of the device pins connected to the signal.
- a listing of the devices making up the circuit, with each device followed by a list of the signals attached to each pin. This type is discussed in a later section entitled "SPICE-type netlists".

The first form (which we will refer to as a signal list) is generated by the $SIGNALS listing command, the latter using the $DEVICES command. Thus, just about the simplest form file that will produce a netlist consists of just one line, as follows:

```
$SIGNALS$SIGNAME $PINS
```

The keyword $SIGNALS specifies that a listing of the signals in the circuit is to be produced, but does not in itself specify the format of the list. The text on the balance of the line determines the format of each signal item. In this example, each line will consist of the signal name ($SIGNAME), followed by a blank, followed by a list of the pin connections associated with this signal. The format of each pin item will be in the default format "device-pin#". This can be overridden using the $SIGPINFORMAT definition statement.

The output file generated from this file and the circuit in Fig. 1 will be as follows:

```
unnamed U1-7 U1-10 U2-2
unnamed U1-2 U2-4
D0 J1-1 U1-14
D1 J1-2 U1-13
D2 J1-3 U1-12
D3 J1-4 U1-11
CLR/ J1-5 U1-1
CLK/ J1-6 U2-3
CEN/ J1-7 U2-1
Ground C1-? C2-? J1-8 U1-3 U1-4 U1-5 U1-6
+5V C1-? C2-? J1-9 U1-9
```

Note the following problems in this netlist:

- several unnamed signals are shown with the same label "unnamed".
- discrete component pins (e.g. C1, C2) are listed with a "?" for the pin number, since no pin number was given on the diagram.
- power and ground pins are not appearing.

The following form file addresses these problems and implements some additional features:

```
$SIGSOURCE(Ground)
$SIGSOURCE(+5V) &Power
$AUTONUMBER(3)
$ASSIGNNAMES $SIGNALS
$SORT $SIGNALS $SIGNAME
$SIGNALS$SIGNAME$COL(15)$PINS
```

This produces the following output file:

```
+5V             C1-2 C2-2 J1-9 U1-9 U1-16 U2-14
CEN/            J1-7 U2-1
CLK/            J1-6 U2-3
CLR/            J1-5 U1-1
D0              J1-1 U1-14
D1              J1-2 U1-13
D2              J1-3 U1-12
D3              J1-4 U1-11
Ground          C1-1 C2-1 J1-8 U1-3 U1-4 U1-5 U1-6 U1-8 U2-7
S00001          U1-7 U1-10 U2-2
S00002          U1-2 U2-4
```

Note the following features of this report:

- the $SORT line causes the nets to be sorted by signal name.
- the $ASSIGNNAMES line causes default names of the form S00001 to be assigned to any signals that are unnamed.
- the $AUTONUMBER format command assigns pin numbers to any device with up to 3 pins.
- the $SIGSOURCE lines indicate that power and ground connections should be extracted from attribute fields of the given names in each device. Connections for the Ground net are extracted from the attribute field with the same name. Connections for the +5V net are extracted from the Power attribute field.
- the $COL keyword is used in the $SIGNALS line to set column alignment and improve readability. Note that this will only appear correctly on the screen if displayed with a fixed-space font such as Courier or Monaco. Another alternative is to put a Tab character in the file at that point, allowing tab settings to be used in a word processor to format the output.

Basic Bills of Materials

The simplest list of the devices making up a circuit is generated by the following single-line form file:

```
$DEVICES$DEVNAME
```

Given the sample circuit in Fig. 1, this will generate the following list:

```
U1
U2
U2
J1
C1
C2
J1
J1
```

Note that there is no particular ordering to this list and like-named devices are not combined. The first problem is fixed by using the $SORT command, as follows:

```
$SORT $DEVICES $DEVNAME
$DEVICES$DEVNAME
```

This generates:

```
C1
C2
J1
J1
J1
U1
U2
U2
```

Like-named devices can be combined into a single report line by inserting the command $COMBDEVSON. The following form also illustrates how attribute parameters can be displayed on each line in fixed columns:

```
$COMBDEVSON
$SORT $DEVICES $DEVNAME
$DEVICES$DEVNAME$COL(20)&Part$COL(40)&Value
```

This produces the following output file:

```
C1          Tantalum      0.1µF
C2          Tantalum      0.1µF
J1          Molex 9-pin
U1          74HCT163
U2          74HCT04
```

IMPORTANT: Note that the COMBDEVSON flag does not actually combine devices with the same name, but rather devices with the same sort position. Since we sorted by device name, this has the desired effect. This can be used to combine lines based on the contents of an attribute field such as the "Part" entry. This is demonstrated in the following form file, which also adds column titles and a use count for each type.

```
$COMBDEVSON
$SORT $DEVICES &Part
Device Type$COL(20)Num.
Used$COL(36)Value$COL(50)References
$DEVICES&Part$COL(25)$COUNT$COL(36)&Value$COL(50)$DEVNAME
```

Note that we are now sorting by type, so COMBDEVSON means, in effect, "combine devices with the same 'Part' field into one line". The $DEVNAME entry here may be ambiguous, since we may be combining devices with different device names onto a single line. If this happens, the report generator will list all values of DEVNAME associated with the "Part" value on that line, as follows:

```
Device Type Num. Used   Value References

74HCT04     1           U2
74HCT163    1           U1
Molex 9-pin 1           J1
Tantalum    2     0.1µF C1 C2
```

IMPORTANT: Since we are sorting by Part value, and not by name, it is important that like-named devices have the same attribute values, otherwise they will appear on different lines.

Spreadsheets and Word Processors

DesignWorks reports can readily be exported to spreadsheet programs such as Microsoft Excel or word processors such as Microsoft Word to take advantage of those programs' formatting and data analysis capabilities. Typical applications include:

- creating formatted copy for presentations or permanent documentation.
- using spreadsheets to compute pricing, board area, power usage, or other parameters of a design.

Export Format

In general, the important element in exporting data to either type of package is *tabs*. Spreadsheets typically use tab characters to separate fields when importing text data, and word processors can make use of tabs to provide adjustable column formatting. Use of blanks and the $COL operator in a form file should be avoided, since blanks do not have a fixed width in most Macintosh fonts and will result in inconsistent results.

Inserting Tabs

A tab character can be inserted anywhere in the report form file and will be passed directly to the output file. Consider the following sample component list generator (based on the last example in the previous section):

```
$COMBDEVSON
$SORT $DEVICES &Part
Device TypetabNum. UsedtabValuetabReferences

$DEVICES&ParttabCOUNTtab&ValuetabDEVNAME
```

Note: The notation tab indicates a tab character entered using the "tab" key on the keyboard.

This format allows us to make use of the tab options in the word processor to improve the look of the report, as follows:

Device Type	Num. Used	Value	References
74HCT04	1		U2
74HCT163	1		U1
Molex 9-pin	1		J1
Tantalum	2	0.1µF	C1 C2

In cases where a single report form command causes multiple items to be placed in a report, such as $PINS, the $ITEMSEPARATOR specification must be used to insert tabs between items in the list. Thus exporting a netlist with tab separators can be done with the following report form (adapted from the preceding section on netlists):

```
$SIGSOURCE(Ground)
$SIGSOURCE(+5V) &Power
$ITEMSEPARATOR(tab)
$AUTONUMBER(3)
$ASSIGNNAMES $SIGNALS
$SORT $SIGNALS $SIGNAME
$SIGNALS$SIGNAMEtab$PINS
```

Page Separators

When exporting data to a spreadsheet, you will normally not want any page breaks to be inserted in the data since they would confuse the importing program. Page breaks can be disabled by setting the $LINESUSED and $PAGELENGTH format options to large values, as follows:

```
$LINESUSED(32000)
$PAGELENGTH(32000)
```

When exporting to a word processor, you can chose to use either of two methods:

- use the above settings and generate a continuous list, allowing the word processor to determine the page breaks.
- set $PAGELENGTH(0) and $LINESUSED(50) (or whatever number suits your application) to force the report generator to insert a form feed character (page break) and a page header on each page.

Other Report Generation Examples

This section illustrates how the Report Generator's sorting and formatting features can be used to extract a variety of useful information from a circuit.

Error Checking Reports

The Report generator can be used to check for various kinds of circuit errors such as missing or overlapping names, missing attribute fields, unconnected signals, etc. Making use of the CHECK command, it is possible to create a form file that checks for these errors without even generating an output file.

Missing Attribute Fields

The following report form will check a circuit for any devices missing the field "SUPPLIER" and write out a list of the offending items:

```
$FIND $DEVICES $NOT &SUPPLIER
$SORT $DEVICES $DEVNAME
$DEVICES$DEVNAME
```

By replacing the $DEVICES command with $CHECK you can have it report the errors immediately instead of writing them to a file:

```
$FIND $DEVICES $NOT &SUPPLIER
$SORT $DEVICES $DEVNAME
$CHECK(Devices missing field SUPPLIER!) $DEVICES
```

Unconnected Signals

The following form file will create a listing of all signals with zero or one pin connections on them:

```
$FIND $SIGNALS $NOT $NUMPINS(2)
$SORT $SIGNALS $SIGNAME
$SIGNALS$SIGNAME
```

Using the Report Generator to Assign Names

A report form file can be used to assign names to groups of devices in a circuit without actually generating a report. For example, the following form file assigns a name of the form "Cn" to all capacitors in the circuit:

```
$FIND $DEVICES $TYPENAME(Cap*) $NOT $DEVNAME
$SORT $DEVICES $PAGE $POSY $POSX
$ASSIGNNAMES $DEVICES $FORMAT(C1)
```

We first find all devices with a library type name beginning with the letters "Cap" and not already named. We then sort the devices by page number and position on the drawing. Then we assign the default names. These names will initially be invisible, but can be made visible by using the Browser tool.

Giving Priority to Nets

When a netlist is to be used to drive a PCB layout system, it is frequently desirable to bring certain nets to the front of the list to give them higher routing priority. This can be done by inserting an attribute field into the highest priority nets and sorting first on this field. Nets will appear in

ascending order of the value in this field. Nets without any field specified will appear last.

In the following example, the field

```
Field     PRIORITY
Value     0
```

has been inserted in the attributes for the net Ground in the sample circuit used in the first sections of this chapter. A reference is made to this field in the $SORT command in the form file, as follows:

```
$SIGSOURCE(Ground)
$SIGSOURCE(+5V) &Power
$AUTONUMBER(3)
$ASSIGNNAMES $SIGNALS
$SORT $SIGNALS &PRIORITY $SIGNAME
$SIGNALS$SIGNAME$COL(15)$PINS
```

This gives the following results:

```
Ground          C1-1 C2-1 J1-8 U1-3 U1-4 U1-5 U1-6 U1-8
+5V             C1-2 C2-2 J1-9 U1-9 U1-16
CEN/            J1-7 U2-1
CLK/            J1-6 U2-3
CLR/            J1-5 U1-1
D0              J1-1 U1-14
D1              J1-2 U1-13
D2              J1-3 U1-12
D3              J1-4 U1-11
S00001          U1-7 U1-10 U2-2
S00002          U1-2 U2-4
```

Note that the Ground net is now listed before all others and the remaining items are still sorted by name.

Devices by page number

This example uses a sort by attribute fields to make an index to locate types used in a design. This is an extract from a multi-page circuit in which each device has a field called "type" specifying its vendor type number. In addition, discrete components have a field called "Value" with their component value. The following report form is used:

```
$COMBDEVSON
$SORT $DEVICES &Part &Value
TYPEtabVALUEtabPAGES

$DEVICES&PartTab&ValueTab$PAGE
```

This form file produces the following results:

```
TYPE        VALUE          PAGES

74HC04                     1 2 3
74HC193                    1 3
RES 1/4W 10K               1 2
RES 1/4W 20K               1
```

Industry Netlist Formats

In this section we will illustrate how the various Report options can be combined to produce standard netlist formats to interface with outside PCB layout, simulation and other design programs on the Macintosh or other machines.

IMPORTANT: The following examples are provided as illustrations only and may not include all the format information needed to reliably interface with a given external system. We have no control over the specifications for third-party systems and these may change from time to time. The standard form files provided on disk with DesignWorks may differ from the examples given here. Use a text editor to view these files directly for more complete and up-to-date information.

SPICE-type Netlists

SPICE-based analog simulation programs read all their circuit structure and simulation information from a text file, which must include the following:

- a title line.
- setup commands which set parameters for the simulation, with one command per line. These usually begin with a ".", such as ".tran" or ".ac".
- one line of information per device in the circuit. Each line starts with the device name (the first letter of which determines its type), a list of the nodes attached to that device, then any parameters required for the device.
- an end marker line.

The order of these lines in the file is not significant, except that the first line is the title line, and the last line must be ".end".

A simplified SPICE report form file contains the following commands:

```
$CREATEREPORT($DESIGNNAME.IN)
$ASSIGNNAMES $SIGNALS $FORMAT(S00000)
$DESIGNNAME $DATE $TIME
&Spice
$DEVICES$DEVNAME $PINS &Spice
.end
```

These commands perform the following functions:

`$CREATEREPORT($DESIGNNAME.IN)` - causes the string ".IN" to be added to the name of the design file to create the default name for the output file.

`$ASSIGNNAMES $SIGNALS` assigns default names to any unnamed signal nets in the circuit.

`$DESIGNNAME $DATE $TIME` - create a title line in the output file consisting of the circuit file name and the date and time at which the report was generated.

`&Spice` - copies the contents of an attribute field called Spice from the design attributes (set up using Set Design Attributes command in the Options menu). This is used to insert SPICE parameter lines before the actual netlist.

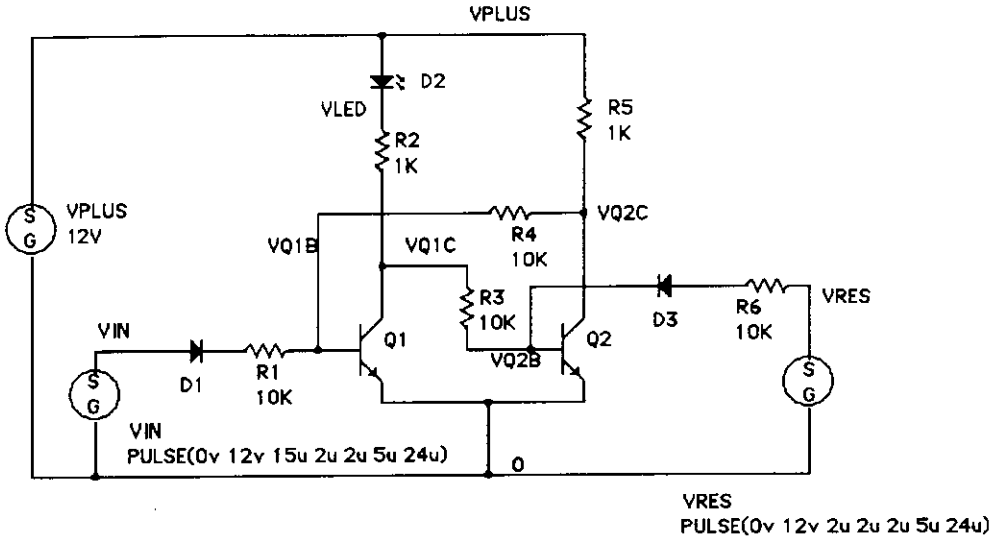
`$DEVICES$DEVNAME $PINS &Spice` - causes a list of devices to be written into the file. Each line will contain the device name, a list of the attached signals (nodes), followed by the contents of an attribute field called Spice. Note that SPICE uses the first letter of the device name to determine its type. It is up to the user to name each device appropriately.

See Chapter VIII - Before Starting a Major Design for information on automatic name generation for SPICE designs.

The $PINS item lists the pin signals in the order that the device pins were defined in the DevEditor. Therefore it is very important that the device libraries used to create the circuit have a pin order that matches SPICE's requirements. The SPICE Devices library provided with DesignWorks meets this requirement.

`.end` - inserts the actual text ".end" in the output file, which is the SPICE termination marker.

Given the following circuit:



the above report form generates the following output file:

```
SPICE Demo  Monday, August 10, 1992  11:05 AM
.TRAN .5US 26US UIC
.model MOD1 npn(bf=80 rb=100 ccs=2pf tf=0.3ns
tr=6ns cje=3pf cjc=2pf
+   va=50)
.model DIODE1 D
Q1  VQ1C  VQ1B  0 MOD1
R1  S00001  VQ1B 10K
R2  VQ1C  VLED 1K
R5  VQ2C  VPLUS 1K
R3  VQ2B  VQ1C 10K
Q2  VQ2C  VQ2B  0 MOD1
D2  VLED  VPLUS DIODE1
D1  VIN  S00001 DIODE1
R4  VQ1B  VQ2C 10K
VPLUS  VPLUS  0 12V
VIN  VIN  0 PULSE(0v 12v 15u 2u 2u 5u 24u)
VRES  VRES  0 PULSE(0v 12v 2u 2u 2u 5u 24u)
R6  S00002  VRES 10K
D3  S00002  VQ2B DIODE1
.END
```

Say, for example, you wished to assign arbitrary names to the devices in your circuit without the restriction of using the first letter to determine the device type. This could be done by placing a field in the attributes for each device containing only the type letter, as in the following example:

```
Field     Value

SPTYPE    V
Spice     AC 1
```

The form file could then be modified as follows to insert this new parameter in the netlist:

```
$DEVICES&SPTYPE$DEVNAME $PINS &Spice
```

The SPICE name would then consist of the concatenation of the contents of the SPTYPE field and the device name. As an alternative to specifying SPTYPE in every device, this field could be entered into the device library entry for this type.

PCAD Format

A PCAD-format (PDIF) netlist for the sample circuit given at the beginning of the Examples section would look like the following:

```
% Project Report Gen Circuit Sunday, March 11,
1992
{COMPONENT Report Gen Circuit.PCB {ENVIRONMENT
Report Gen Circuit} {DETAIL {SUBCOMP
{I Tantalum C1
{CN 1 Ground 2 +5V
}
}
{I Tantalum C2
{CN 1 Ground 2 +5V
}
}
{I Molex 9-pin J1
{CN 1 D0 2 D1 3 D2 4 D3 5 CLR/ 6 CLK/ 7 CEN/ 8
Ground 9 +5V
}
}
{I 74HCT163 U1
{CN 1 CLR/ 2 SIG3 3 Ground 4 Ground 5 Ground 6
Ground 7 SIG1 8 Ground 9 +5V 10 SIG1 11 D3 12 D2
13 D1 14 D0 15 SIG2 16 +5V
}
}
{I 74HCT04 U2
{CN 1 CEN/ 2 SIG1 3 CLK/ 4 SIG3 7 Ground 14 +5V
}
}
}}}
```

Note that this is a listing by device, not by signal, similar to the SPICE form illustrated previously. Each new device is introduced by the "{I" then the type, name, and list of attached signals follows.

This was generated using the following form file:

```
$CREATEREPORT($DESIGNNAME.PCAD) $PROMPT
$SIGSOURCE(Ground)
$SIGSOURCE(+5V) &Power
$SIGSOURCE(+12V) &Plus12V
$SIGSOURCE(-12V) &Minus12V
$AUTONUMBER(3)
$COMBDEVSON
$ASSIGNNAMES $DEVICES $FORMAT(U1)
$ASSIGNNAMES $SIGNALS $FORMAT(SIG1)
$SORT $DEVICES $DEVNAME
$DEVPINFORMAT $PINNUM $SIGNAME
% Project $DESIGNNAME $DATE
\{COMPONENT $DESIGNNAME.PCB \{ENVIRONMENT $DESIGNNAME\} \{DETAIL
\{SUBCOMP
$DEVICES\{I &Part $DEVNAME$NEWLINE\{CN $PINS$NEWLINE\}$NEWLINE\}
\}\}\}
```

Note the following important features of this form file:

- the CREATEREPORT command causes the default report file name to be set to the name of the design with ".PCAD" appended. The user will be prompted to verify or change the name before proceeding.
- the SIGSOURCE commands provide the names of power and ground nets that can be specified in device attribute fields. Note that in the above sample netlist, several power and ground connections are shown that do not appear on the diagram. These are derived from device attributes. Every device in the circuit will be searched for an attribute field with any of the given names. If a field is found, the list of pin numbers it contains will be added to the net with the same name. Note that it is possible to use an attribute field with a different name than the net, if needed. See the SIGSOURCE command for details.
- the AUTONUMBER command specifies that devices with 3 or less pins will have pin numbers assigned automatically even if they do not appear on the diagram. This is intended to number discrete components which do not have numbers showing on the diagram.
- the COMBDEVSON command specifies that multiple devices with the same sort value (in this case, the same device name) should be combined into a single device entry.
- the ASSIGNNAMES commands cause default names to be created for any unnamed signals or devices. For devices, the names will be of the

form "Un" where "n" is an integer. For signals, the names will be of the form "SIGn". The circuit is searched before assigning names to ensure that the new names do not match any existing ones.

- the SORT command sorts the devices by name, for improved readability.

- the DEVPINFORMAT command sets the format for each device pin in the following $DEVICES listing. In this case, each entry will show the pin number followed by a blank and the signal name.

- the "% Project" text is not a command, so it is transmitted directly to the output file. The $DESIGNNAME command is replaced by the name of the circuit being reported. The $DATE command is replaced by the date the report was created.

- in order to meet the PCAD format requirements, the "COMPONENT" line must start with an opening brace "{". Since this is the Report Generator's comment character, it must be preceded by an escape character "\". The backslash character does not appear in the output. The rest of this line is straight text, except for the $DESIGNNAME commands that are substituted as noted above.

- the DEVICES line is the heart of the report. Everything following the "$DEVICES" keyword on the line will be repeated once for each device listed. In this case, each new device will be introduced by the text "{I " (note the escape character required for each brace), followed by the type name (extracted from an attribute field), the device name, then a carriage return (NEWLINE). The second line of the entry will contain the text "{CN " followed by a list of the pins attached to the device. Each device entry will be terminated by two closing braces, matching the opening ones at the beginning.

- the whole file is terminated by three closing braces.

Draftsman EE Format

The format used by the Draftsman EE printed circuit board layout package is a listing by signal, as follows:

```
NET 1 +5V          1          C1.2
          2        C2.2
          3        J1.9
          4        U1.9
          5        U1.16
          6        U2.14
NET 2 CEN/         1          J1.7
          2        U2.1
NET 3 CLK/         1          J1.6
          2        U2.3
```

```
NET 4 CLR/          1           J1.5
          2         U1.1
NET 5 D0  1         J1.1
          2         U1.14
NET 6 D1  1         J1.2
          2         U1.13
NET 7 D2  1         J1.3
          2         U1.12
NET 8 D3  1         J1.4
          2         U1.11
NET 9 Ground        1           C1.1
          2         C2.1
          3         J1.8
          4         U1.3
          5         U1.4
          6         U1.5
          7         U1.6
          8         U1.8
          9         U2.7
NET 10 SIG1         1           U1.7
          2         U1.10
          3         U2.2
NET 11 SIG2         1           U1.2
          2         U2.4
```

This is generated using the following form file:

```
$CREATEREPORT($DESIGNNAME.DEE) $PROMPT
$ASSIGNNAMES $SIGNALS $FORMAT(SIG0)
$ASSIGNNAMES $DEVICES $FORMAT(DEV0)
$SORT $SIGNALS $SIGNAME
$SORT $DEVICES $DEVNAME
$CONTSTART(tab)
$MAXITEMSPERLINE(1)
$ITEMSEPARATOR(tab)
$SIGPINFORMAT $PINSEQ(1)tab$DEVNAME.$PINNUM
$SIGNALSNET $SIGSEQ(1) $BUSNAME(.)$SIGNAMEtab$PINS
```

Note the following unique features of this form file:

- the ASSIGNNAMES commands will generate names with the given default format.
- only one pin entry is placed on each line and each line starts with a tab character.
- all nets and pin entries must be numbered sequentially. This is done using the $PINSEQ and $SIGSEQ sub-commands with a given number starting point.
- the BUSNAME sub-command is used to prefix signal names with the name of the bus containing them, if any.

Cadnetix Format

The following form file generates a netlist appropriate for a Cadnetix PCB layout system:

```
$CREATEREPORT($DESIGNNAME.Cadnetix) $PROMPT
$ROWS(60)
$COLUMNS(80)
$SIGSOURCE(Ground)
$SIGSOURCE(+5V) &Power
$AUTONUMBER(3)
$BLANKREPLACE(_)
$COMBDEVSON
$PINFORMAT $DEVNAME $PINNUM
$CONTEND( $)
$ASSIGNNAMES $SIGNALS
$ASSIGNNAMES $DEVICES
$SORT $SIGNALS $SIGNAME
COPYRIGHT  1993 &Company
DESIGNERS: &Designers
DATE:      $DATE
FILE:      $DESIGNNAME.Cadnetix
PARTS LIST
ID &ProjectID
$SORT $DEVICES &Part
$DEVICES&Part &Package $DEVNAME
EOS
$MAXITEMSPERLINE(8)
$SORT $DEVICES $DEVNAME
NET LIST
$SIGNALSNN $SIGNAME $PINS
EOS
```

Note the following unique features of this format:

- the BLANKREPLACE command is used to ensure that blank characters do not appear in any names in the file.
- the CONTEND command sets the string " $" to be placed at the end of any data line that is being continued to the next line because the LINEWIDTH or MAXITEMSPERLINE settings were exceeded.
- the ASSIGNNAMES commands assign default names to any objects that were unnamed on the schematic. The default format of "D0001" for devices and "S0001" for signals will be used since no format is specified.
- a number of design attribute fields are referred to, including &Company, &Designers, and &ProjectID. These must be specified in

the Set Design Attributes command or they will not appear in the report file.

- the MAXITEMSPERLINE command is used to specify how many pin entries can appear on a single line before a continuation line is created.
- the Report Generator stops scanning for a keyword once it recognizes something. For example the line "$SIGNALSNN" is interpreted as the keyword "$SIGNALS" followed by the literal text "NN". This does not apply to attribute field names, since these are not predefined. They must be terminated by a blank or special character.

Report Form Language Reference

This section of the manual provides detailed information on the creation of report form files. It is not necessary to read this section in order to use the standard form files provided. In order to get a quick overview of the capabilities available, it is suggested that you review the Examples section first.

Overview

The form file required to create a report consists of a series of commands, variables and text, referred to as the Form Language. A form file can be created using any text editor or word processor. If you use a word processor, be sure to select a "Text Only" option when saving the file.

A number of standard form files are provided with DesignWorks and many of these are explained in the Examples section following. These can be used as guides in creating your own form files.

DesignWorks Object Types

For report generation purposes, a DesignWorks circuit consists of two types of objects: devices and signals. The Report tool has two general sets of options allowing information on each of these types of objects to be written to a file in a variety of forms.

Devices

The device listing features are used to create component lists, bills of materials and types of netlists that are listed by device, such as a SPICE or PCad format. For each device in a list, any of the following data can be included:

- Device Name - this is the name applied to the device instance, e.g. "U1", "R23", "J12", etc.
- Device Attributes - the contents of any device attribute fields can be included in a listing.
- Device Token - the device token is an integer that is assigned automatically by DesignWorks when the device is first placed on the diagram and not changed as long as that device exists in that circuit. This token number can be used by an external program to determine

which devices are new and which are old in a circuit, despite name or parameter changes.

• Device Sequence Number - this is a temporary integer that is assigned by the Report tool as part of its sorting process and may change during a report if multiple sorts are done. This can be used in cases where external programs require objects to be sequentially numbered.

• Device Type information - this refers to information retrieved from the library when the device was created, such as the type name (e.g. "74LS123"), number and types of pins, etc.

• Device Graphical Information - information about the device's page number, position and orientation on the schematic are available to allow transfer of schematic data to other systems, or to create reference listings.

• Attached signals - a device listing can also contain a list of the signals attached to each pin of the device. Each signal entry can reference any of the signal data described below. See the $DEVPINFORMAT command.

Device listings are created using the $DEVICES command, described later in this chapter.

Signals

Signal listings are used primarily for creating netlists for transfer to external simulators or PCB systems. The following information is available about each signal in the circuit:

• Signal Name - this is the name of the signal as it appears on the schematic, or is assigned automatically by the report generator.

• Signal Attributes - the contents of any signal attribute field can be included in a signal listing.

• Signal Token - this is an integer that is assigned automatically by DesignWorks when the signal is first created and not changed as long as that signal exists in that circuit. Note however, that many common editing operations result in signals being merged or broken apart, which will often result in the signal token changing even though some of the connections in the old signal remain. This token number can be used by an external program to determine which signals are new and which are old in a circuit, despite name or parameter changes.

• Signal Sequence Number - this is a temporary integer that is assigned by the Report tool as part of its sorting process and may change during a report if multiple sorts are done. This can be used in cases where external programs require objects to be sequentially numbered.

- Signal Graphical Data - information about page numbers and coordinates of attached lines on the schematic can be included in a signals listing.
- Attached device pins - any signal list can contain a list of the device pins interconnected by this signal. Pin format options allow almost any of the device parameters mentioned above to be included in each pin entry. See the $SIGPINFORMAT command.

Signals listings are created using the $SIGNALS command described later in this chapter.

Form File Structure

When the Report command is selected by the user, the form file is opened and read sequentially. Any text that is not recognized as a valid command, variable or comment is written directly to the output file unchanged. When a command or variable is found, text generated by the system is substituted for the command or variable text. In some cases, such as when a signal or device list is called for, the line containing the command will be repeated once for each item in the list.

Any number of listing commands can be given in one file. Thus, a single output file can contain a number of netlists and component lists in a variety of formats, if needed.

The form file contains four types of data:

1) **Commands**: commands start with the "$" symbol and are in capital letters. Commands are used to control the format of the report. Some of the commands need parameters to set certain values (e.g. the number of characters on a line). Any parameters are enclosed in parentheses following the command word. If the desired output file format requires that a word starting with a "$" be placed in the form file, it is best to precede the "$" with the escape character "\" (backslash) to ensure that it will not be accidentally interpreted as a command.

Commands fall into the following general sub-groups:

- definition commands are those which set parameters for how data is generated but do not themselves generate any text in the output file. An example is the $SIGPINFORMAT command, which specifies how pins will be displayed in a signal list.
- listing commands are those which cause lists of items to be directed to the output file, for example the $SIGNALS command, which generates a list of the signals in the circuit. All the commands which follow a

listing command on a line are item-specific, i.e. their exact meaning depends upon the type of item being listed.

- item-specific commands are those which can appear after a listing command and whose meaning depends upon the type of item being listed. For example, the $EVENTS command can appear in a $SIGNALS listing only and causes a list of timing data for that signal to be produced. The $EVENTS keyword has no meaning in any other part of the file.

- system variables are commands that cause a specific internal value to be directed to the output file, e.g. the circuit file name or the system date.

2) **Attribute variables**: Attribute variables start with the "&" symbol and are used for handling the user-defined attribute fields in devices, signals, or the circuit itself. The variable will be replaced with the actual value of that field (found in each item). For example &Pwr might appear as 140mW in the report for one of the devices in the circuit.

If the desired output file format requires that a "&" be placed in the form file, it is best to precede the "&" with the escape character "\" (backslash) to ensure that it will not be accidentally interpreted as an attribute variable.

3) **Comments**: comments are messages enclosed between the left and right braces "{" and "}". Comments are intended for internal documentation of the command file. No characters between or including the braces, or a carriage return following the closing brace, will be transmitted to the report. If you need to transmit the character "{" to the output file, you must precede it by the "escape" character "\" (backslash) so that it will not be recognized as a comment.

4) **Data:** All other text and special characters (e.g. tabs and carriage returns) found in the command file are considered data and are transmitted to the output file unmodified. Any item starting with a $ or & but not recognized as a command or variable word will be transmitted to the report file. No error messages are generated for incorrect keywords. It is not recommended that Macintosh "option key" characters be used in reports since they use character codes outside the normal ASCII character set. This may result in difficulty transmitting the files to other programs and some of the Report format options may not function correctly.

Definition Commands

The definition commands control the general format of the report, for example:

1) the number of rows per page.

2) the number of columns per row.

3) the specification of an attribute field as a signal source (e.g. for ground or power connections).

4) other report formatting options

Each of these parameters is set by using specific command words, as indicated below.

Page Format

$LINEWIDTH(n)

The LINEWIDTH command specifies the maximum number of characters that will be placed on a line before an automatic carriage return is inserted. See the MAXITEMSPERLINE command below, which can be used to specify the length as an item count instead of a character count. The CONTEND and CONTSTART commands can be used to indicate how continuation lines are implemented.

The default value for LINEWIDTH is 255.

$PAGELENGTH(n)

The PAGELENGTH command is used to specify the action to be taken when a page is full (i.e. when the number of lines specified in LINESUSED is exceeded). If PAGELENGTH is set to 0, then a Form Feed character (ASCII code 12) is placed in the file. This character is accepted by many word processors as a page break. If PAGELENGTH is set to any other integer, then that number is taken to be the total number of lines from the top of one page to the top of the next. To disable all page breaks, set PAGELENGTH and LINESUSED to very large values (e.g. 32,767). The default value is 66.

$LINESUSED(n)

The LINESUSED command specifies number of lines to use on each report page before moving to the next page. The actual length of the page is

specified by PAGELENGTH above. The default value is 32767, i.e. page breaks are disabled.

$MAXITEMSPERLINE(n)

The MAXITEMSPERLINE command is used to restrict the number of entries placed on a line in repeating structures such as a netlist. The default value is 32,767.

Common Signal Connections

The "signal source" facility allows you to name certain signals to be treated as power and ground nets. This has the following two effects:

- It allows you to create common connections in a circuit using attribute entries in devices. This can be used to create power and ground entries in the netlist without having to show all these connections explicitly on the diagram.
- It informs the Report generator that the given signals should be merged across hierarchy levels.

See more information on power and ground nets in hierarchy later in this chapter.

Any device attribute field may be specified as a signal source by the command:

```
$SIGSOURCE(signalName) &fieldName
```

"signalName" is the name of the signal to attach pins to in the netlist. "fieldName" is the name of the attribute field to search for in each device to look for pin numbers to attach to the named signal. If &fieldName is omitted, the signal name is taken to be the name of the attribute field to search.

The named device attribute field is assumed to contain a list of pin numbers to attach to the signal. This can consist of a single pin entry, such as "7", or a list, such as "5,6,9,14".

For example "Ground" connections can be created as follows:

- Using the Get Info command on a selected device, create an entry such as "7" in the attributes for a device, in this case connecting pin 7 to the Ground net.

• Place the command $SIGSOURCE(Ground) in the form file. This causes Report to search all device attributes for fields named Ground and use the field value as a pin number.

There is no fixed limit to the number of SIGSOURCE entries that can be created.

NOTE: The predefined fields Ground and Power are normally used for standard power and ground connections. These pin connections are prespecified for all digital components in the standard DesignWorks libraries. Corresponding $SIGSOURCE statements are included, where appropriate, in all standard Report form files. You can create your own special-purpose power nets by using the Define Attribute Fields command to define a new field, then adding the appropriate SIGSOURCE statement to the form file.

Auto-numbering Discrete Components

Automatic numbering of device pins can be specified by the command:

```
$AUTONUMBER(number of pins)
```

Any device with less than or equal to the number of pins specified will have pin numbers automatically assigned if none were present in the circuit file. This option is intended to provide pin numbers for discrete components in a circuit, since they do not normally have pin numbers on a diagram. Accordingly, a typical use of this command would be AUTONUMBER(3). The default number is zero.

Setting Pin Order in Device Lists

Some design programs, such as SPICE and SILOS, require netlist formats in which the device pin order is significant. Since there are no standards for pin order, different packages will likely require different ordering schemes. In order to allow a variety of pin orderings to be stored with each device type, the Report tool can be asked to search a device attribute field for pin order information. The name of this field is specified in the DEVPINSEQUENCE command, as follows:

```
$DEVPINSEQUENCE &attrField
```

Once this field has been specified, if a $PINS sub-command is encountered in a $DEVICES listing, each device is searched for an attribute field with the given name. If the field is found (in the device instance or the type attributes), then it is assumed to contain a list of names of the pins to write

out to the file. The pin names must be listed exactly as they appear in the DevEditor's pin list, or, in the case of SIGSOURCE pins, as the fields are named in the device attributes. If the specified field is not found in a given device, then the normal pin ordering is used.

For example, assume a 2-input gate has pins named A, B and OUT. A pin sequence attribute field could be specified in the device (or in the library type attributes) as follows:

```
Field     SPICEPINS
Value     OUT,A,B
```

The command:

```
$DEVPINSEQUENCE &PinSequence
```

in the form file would then cause subsequent listings of that device to have pins appear in the specified order. The predefined field PinSequence is intended for this purpose.

Setting Report Hierarchy Mode

The command `$HIERARCHY` is used to enable hierarchical netlisting and set the type of processing that will be done in generating a file from a hierarchical design. If this command does not appear in a file, then the default mode will be CIRCUIT, i.e. the contents of the current circuit level only will be listed. Example:

```
$HIERARCHY $FLAT
```

The HIERARCHY command must appear before any ASSIGNNAMES, FIND, SORT, DEVICES or SIGNALS commands to ensure that the internal devices are included in these operations.

See more information in the section on Hierarchical Net Listing below and in the Report Examples section.

Format Options

$COMBPINSON/$COMBPINSOFF

When ON causes multiple pin connections on the same device to appear, for example, as:

```
IC1-2,5,6,12
```

instead of:

```
IC1-2  IC1-5  IC1-6  IC1-12
```

Default is OFF.

$ALIGNCOLSON/$ALIGNCOLSOFF

When ON causes extra blanks to be inserted between netlist or component list entries to force column alignment. This improves readability when used with a fixed-space font such as Monaco or Courier. If ALIGNCOLS is OFF, then the item separator will be a tab if TABFIELDS is ON, or a single blank otherwise. Default is OFF. NOTE: You may have to insert a number of blanks at the front of continuation lines using CONTSTART to force correct alignment of the first item in each continuation line.

$SPACE(number)

The SPACE item sets the column spacing used when ALIGNCOLSON is set. Default is 16. E.g:

```
$SPACE(32)
```

sets the column spacing to 32.

$ITEMSEPARATOR(string)

The ITEMSEPARATOR command specifies a string that will be inserted between successive items in an item list. If ITEMSEPARATOR is not specified, then item spacing will be determined by TABFIELDS and ALIGNCOLS. If it is specified, it will override TABFIELDS and ALIGNCOLS. Example:

```
$ITEMSEPARATOR(,)
```

This example will insert a comma between items in a list.

$UNNAMEDDEVS(string)

The UNNAMEDDEVS command specifies a string that will be output whenever a device name is called for and the device in question is unnamed. For example:

```
$UNNAMEDDEVS(?????)
```

$UNNAMEDSIGS(string)

The UNNAMEDSIGS command specifies a string that will be output whenever a signal name is called for and the signal in question is unnamed. The default value is "unnamed". For example:

```
$UNNAMEDSIGS(?????)
```

$ONEPINSON/$ONEPINSOFF

When ON suppresses the pin number from being printed in netlist entries for devices with only one pin. This will cause entries such as test points to appear as TP1 instead of TP1-1. Default is OFF.

$CONTSTART(string)

The CONTSTART item specifies a string of characters that will be inserted at the beginning of each line that is a continuation from a previous line (i.e. whenever MAXITEMSPERLINE or LINEWIDTH is exceeded while writing a list). Examples:

```
$CONTSTART(              )
$CONTSTART($ )
```

The first example will insert a number of blanks in front of each continuation line to set column alignment. The second example inserts a $ sign and a blank to mark continuation lines.

$CONTEND(string)

The CONTEND item specifies a string of characters that will be inserted at the end of each line that is being continued onto the next line (i.e. whenever MAXITEMSPERLINE or LINEWIDTH is exceeded). Example:

```
$CONTEND( >)
```

This example will insert a blank and a > sign at the end of a line when an item list has overflowed the line.

$BLANKREPLACE(string)

The BLANKREPLACE item specifies a string of characters that will be substituted for a blank in any device or signal name. This is done to accommodate systems which cannot accept blanks in names. Examples:

```
$BLANKREPLACE()
$BLANKREPLACE(_)
```

The first example will remove any blanks from names. The second example will replace each blank in a name with an underscore character.

$UNCONNPINSON/$UNCONNPINSOFF

When ON, allows device pins that have no signal lines attached to them and are not connected to any other signals by name to appear in the netlist, otherwise they are suppressed. Default is ON.

$REPEATITEMON/$REPEATITEMOFF

This option, when ON, forces the last item in any list to be repeated at the start of any continuation line. This is used by some systems to recognize continuations.

$CONTREPEATON/$CONTREPEATOFF

This option, when ON, forces the first part (up to the start of the first pin item) of the first line in a signal list to be repeated at the start of any continuation line. This is used by some systems to recognize continuations by repeating the signal name or other parameters on each successive netlist line for the same signal.

$ATTRNUM(number) &attrField

The ATTRNUM command provides a way of placing sequential numbers for devices or signals in a report, while allowing the numbers for specific objects to be set by an attribute field. For example, some netlist formats require that power and ground nets be numbered with specific values, then all other nets sequentially numbered with higher values.

The ATTRNUM command specifies the name of an attribute field to search for in each object. If it is found, the field value is placed in the report, otherwise a sequential number is substituted. The attribute field must also appear somewhere in the relevant $DEVICES or $SIGNALS command line. The substituted number is generated in the same format as it is given in the

ATTRNUM command, and starting with the value given. I.e. leading zeros and other text are retained.

For example:

```
$ATTRNUM(00100) &NETNUM
$SIGNALS&NETNUM $PINS
```

In this case, if a field called "NETNUM" exists in a signal, the contents of that field will be used. Otherwise, a number will be generated having 5 digits and starting at 100 for the first one generated.

File Control Commands

Output File Creation

The command $CREATEREPORT is used to set the report file name and prompt the user to change it if desired. The format of this command is as follows:

```
$CREATEREPORT(reportName) $PROMPT $CREATOR(type)
```

"reportName" can consist of any mix of random text and the variable "$DESIGNNAME" (which inserts the name of the circuit being reported) or an attribute field name &field (which inserts the contents of the specified design attribute field.

The $PROMPT option specifies that the user should be prompted with a standard file creation box with the given name as a default. Otherwise the file is created immediately with no user interaction. If a file already exists by the given name, it is overwritten.

The $CREATOR option is used to set the Macintosh "creator type" of the output file. This determines what application is started when the file is opened and is always exactly 4 characters. It is beyond the scope of this manual to discuss the complete operation of this mechanism, but typical values are "ttxt" to invoke TeachText, "MPS " (the blank is significant) for the MPW programming environment, or "MSWD" for Microsoft Word. The default value is "ttxt".

The following example creates a file with the name of the circuit plus the text " Report" attached and prompts the user to verify or change the name.

```
$CREATEREPORT($DESIGNNAME Report) $PROMPT
```

The following example creates a report named the same as the circuit plus ".IN" and does not prompt the user for confirmation. The file will have a creator type of "MPS ".

```
$CREATEREPORT($DESIGNNAME.IN) $CREATOR(MPS )
```

The following example creates a file called exactly "Output File" with no user confirmation.

```
$CREATEREPORT(Output File)
```

The following looks for a field in the design attributes called "SPICEFILE" and uses the contents of that as the report file name, with no user confirmation.

```
$CREATEREPORT(&SPICEFILE)
```

IMPORTANT: If the CREATEREPORT command appears in the form file after some command that generates output text, then the user will be prompted to create a file with the circuit name plus " Report" and the $CREATEREPORT will be ignored. The best way to ensure correct operation is to put this command at the very beginning of the file. If it does not appear to be working, check for blank lines or spurious characters at the beginning of the file.

$CLOSEREPORT

The CLOSEREPORT command causes the current report output file to be closed. This can be followed by another CREATEREPORT command to create another output file, thus allowing a single form file to generate multiple report files. The output file is normally closed automatically at the end of the form file, so this command is only necessary to create multiple output files. If CLOSEREPORT is not followed by a CREATEREPORT command, and subsequent listing commands attempt to write output data, then the user will be prompted to provide a name for a new output file.

Pin Format Commands

The following format commands specify the way individual pins are presented in reports. Each format command must be followed on the line by a combination of format sub-commands (described below) and text which described the appearance of each item entry. All leading and trailing blanks in the format description are ignored. Any text not recognized as a sub-command is placed in the file as part of each item entry.

$PINTYPEFORMAT string1 string2 ...

This command sets the output string used for each pin type wherever $PINTYPE is specified in a device listing. The strings are given in the following order, separated by blanks:

Item order	Function	Default string
1	Input	INPUT
2	Output	OUTPUT
3	3-state	3STATE
4	Bidirectional	BIDIR
5	Not used	?????
6	Open Collector	OC
7	Undefined	?????
8	Bus	BUS
9	Perm. low	LOW
10	Perm. high	HIGH
11	Latched input	LTCHIN
12	Latched output	LTCHOUT
13	Clocked input	CLKIN
14	Clocked output	CLKOUT
15	Clock input	CLOCK
16	Open Emitter	OE
17	No connection	NC

Note: "Undefined" can only occur if a device was being created in the DevEditor and was closed before all its pins had been specified. If less than 17 strings are specified, the trailing items retain their default value.

For example, to set the pin type string for inputs and outputs (leaving all others at their default values):

```
$PINTYPEFORMAT Inp Out
```

$DEVPINFORMAT/$SIGPINFORMAT

This command sets the format for pin lists appearing in a $DEVICES listing or $SIGNALS listing, respectively. For $DEVICES, the default format is "$SIGNAME", meaning the signal name alone will appear for

each device pin. For $SIGNALS, the default format is "$DEVNAME-$PINNUM", giving a pin item such as "U1-12".

The following format keywords are valid after $DEVPINFORMAT or $SIGPINFORMAT:

$CHILDSIGNAME	The name of the signal attached to this pin in the device's internal circuit, or null string if none. This is intended for creating sub-circuit headers for formats requiring the names of the internal signals attached.
$DEVHIERNAME	The hierarchical name (i.e. prefixed by names of parent devices) of the device.
$DEVINSTNAME	The contents of the InstName field on the device.
$DEVNAME	The name of the current device.
$DEVSEQ(n)	The sequence number of the current device, i.e. an integer assigned to the device, starting at "n" for the first device in the sorted sequence and incremented for each subsequent one. This number is not permanently associated with the device but it strictly its sequence number in the current sort. The "(n)" part is optional. If it is omitted the sequence starts at zero. If no sort has been done, all devices will have sequence number "n". NOTE: DEVSEQ will be zero if no SORT has been done.
$DEVTOKEN	The token number permanently assigned to the device.
$GRID	The page reference grid position (e.g. C3) of the current item.
$NEWLINE	Inserts a line break in the pin entry.
$PAGE	The page number of the current item.

$PINDIR	A two-character code representing the orientation of the current pin. **Output String Direction** E1 East N1 North W1 West S1 South
$PINNAME	The name of the current pin (i.e. as it appears in the DevEditor's pin list window).
$PINNUM	The visible (user-applied) number of the current pin.
$PINSEQ(n)	The sequence number of the current pin, i.e. a sequence number starting at "n" for all pins listed. This can be followed by an optional number in parentheses representing the starting point for numbering. If it is omitted, zero is assumed.
$PINTYPE	A string representing the type of the current pin. The string used for each pin type can be set using the $PINTYPEFORMAT command, described above.
$POSX	The X coordinate of the current pin in 1/1000".
$POSY	The Y coordinate of the current pin in 1/1000".
$SIGHIERNAME	The hierarchical name (i.e. prefixed by the names of parent devices) of the current signal.
$SIGINSTNAME	The contents of the signal's InstName field.
$SIGNAME	The name of the current signal.

$SIGSEQ	The sequence number of the current signal, i.e. an integer assigned to the signal, starting at "n" for the first signal in the sorted sequence and incremented for each subsequent one. This number is not permanently associated with the signal but it strictly its sequence number in the current sort. The "(n)" part is optional. If it is omitted the sequence starts at zero. If no sort has been done, all signals will have sequence number "n".
$SIGTOKEN	The token number assigned to the signal.
$SYSPIN	The system pin number of the current pin, i.e. its ordinal number as it appears in the pin list in the DevEditor, starting at 1.
&attrField	The contents of the given attribute field on the current pin.

Format Notes Section

The $NOTES keyword introduces a special comments area that can be viewed by the user by clicking on the Format Notes button in the Report dialog.

NOTE: We recommend including a $NOTES section in every form file you create indicating the purpose (i.e. output format) of the form, the expected hierarchy mode of the design, and all attribute fields used.

The notes section starts with the $NOTES keyword on a line and ends with a $END keyword on a line. All text on the intervening lines will be ignored when generating a report but will be displayed in the Format Notes box on the screen. Following is an example:

```
$NOTES
This format generates a "flattened" netlist file
for PSpice from a hierarchical or flat design.

Non-standard attribute fields required:

Field name     Where      Usage

Spice          Devices    SPICE parameters

$END
```

Page Header Format

The Header section controls what will appear at the top of each page of the report. This section starts with a $HEADER keyword and ends with $ENDHEADER. Note that the contents of this section is not passed to the output file immediately, but only when a page feed occurs. The Header section command words and their results are given below:

`$PAGE` - Puts out the current page number of the report being generated.

`$DATE` - Puts out the current day, month and year.

`$TIME` - Puts out the current system time.

`$DESIGNNAME` - Puts out the name of the circuit file being used as input for the report.

`$ENDHEADER` - Marks the end of the header section.

`&attr` - if an attribute variable appears anywhere in the header section, then the program attempts to match the variable name with a field in the design attributes. If a match is found, the value of the field is substituted. This is intended to accommodate global items such as titles, circuit parameters, routing instructions or other info that is not related to any one device or signal.

Example page header specification:

```
$HEADER
XYZ Industries of Smithville, Inc.
File: $DESIGNNAME     Date: $DATE
$ENDHEADER
```

Assigning Default Names

```
$ASSIGNNAMES type $FORMAT(pattern) &field $TYPENAME
```

The ASSIGNNAMES command is used to apply names to unnamed devices or signals in the circuit before a listing is generated. If this is not done, any object which has not been named (either manually or by the auto-naming or packaging features) will appear as "unnamed" in a net or component list.

> IMPORTANT NOTES:
>
> 1) The names assigned by ASSIGNNAMES will be invisible but permanently associated with the object. They can be made visible either using the Attributes command associated with the object or the Browser tool.

2) If the report hierarchy mode has been set to $FLAT or $FLATDOWN and the design is in Physical hierarchy mode, then the ASSIGNNAMES command sets the InstName field, not Name.

Note that ASSIGNNAMES can be used in conjunction with the FIND command to assign names to selected subsets of devices, such as all resistors, etc. See the Examples chapter for more information.

The "type" specification (either "$DEVICES" or "$SIGNALS") is mandatory, all other items are optional.

The $TYPENAME option is valid only for devices and specifies that the device type name should be used as part of the name with a numeric part appended.

The $FORMAT item specifies a default format for the assigned names, which can be overridden by the &field item, as explained below. The pattern specified in parentheses after $FORMAT contains a non-numeric part followed by a numeric part (specified as zero digits). The non-numeric part is copied directly to each default name generated. The numeric part is converted to an integer, which is incremented for each name generated. Leading zeroes can be specified by including more than one zero in the numeric part. Example formats:

```
$FORMAT(IC0)
```

will give IC1, IC2, IC3, etc.

```
$FORMAT(U000)
```

will give U001, U002, U003, etc.

If $FORMAT is not specified, format D00000 will be assumed for devices, and S00000 for signals.

The `&field` item allows you to specify an attribute field to search for a string to use as a name prefix. All discrete component libraries supplied with DesignWorks contain a user field called Name.Prefix which specifies the commonly-used letter for the device type, such as "R" for resistors, "C" for capacitors, etc. Thus, if the item &Name.Prefix appears in the ASSIGNNAMES command, resistors will be numbered R1, R2, etc., and capacitors C1, C2, etc.

As with other user field searches, the Report Generator searches in a number of places to find a prefix. For devices the device instance is searched first, then the data in the library entry. This allows you to specify defaults for values which can be overridden for specific cases.

Regardless of how the format is specified, the circuit is searched before any name assignments are done and the non-numeric and numeric parts of each object name are extracted. The numeric parts of any assigned names are generated starting at the next integer higher than any in the circuit.

Examples:

```
$ASSIGNNAMES $DEVICES $FORMAT(U0)
```

This will assign names starting at U1 (or the first "Un" name that is unused). The names will be removed after the report has been generated.

```
$ASSIGNNAMES $SIGNALS
```

This will assign signal names to unnamed signals starting with S00001. The names will be removed from the circuit after the report has been generated.

```
$ASSIGNNAMES $DEVICES &Function
```

This will assign names using the string found in the field Function in each device, or names such as D00001 if such a field is not found or is empty. The assigned names will be kept permanently with the circuit.

Selection and Sorting Commands

The SORT Command

The SORT command provides the ability to sort device and signal listings on any data field. Items with the same value in any field can be optionally merged into a single line. This allows listings to be organized to suit various applications, for example:

- device or signal lists can be sorted by name to enhance readability.
- device lists can be sorted by type so that each line lists one device type with all information common to that type and a list of instances of the type.
- a device list can be sorted by page number to show the devices used on each page.
- signals can be sorted by an attribute field, for example to give priority to certain nets so they are listed first for autorouting purposes.
- devices can be sorted by any attribute field, for example by component value, stock number, cost, etc.

The SORT command has the following form:

```
$SORT objectType field1 field2 ...
```

"objectType" must be either "$DEVICES" or "$SIGNALS".

"field1", "field2", etc. are identifiers indicating which fields to sort on. The list is sorted first on the first field. If any items in the list have identical values in that field, then the groups of like-valued items are sorted on the next specified field, etc.

Once a sort has been done, it remains in effect for all subsequent listings on that object type until the next SORT command. Each SORT clears the previous SORT. Sorting can be in ascending or descending order, as described below.

The following field specifications are valid in all SORT commands:

$ASCENDING	This specifies that all following fields specified should be sorted in ascending order. This is the default direction, so ASCENDING will normally only be required after a DESCENDING to reset the sort direction for subsequent fields on the same line.
$DESCENDING	This specifies that all following fields will be used to sort in descending order.

If "objectType" is $DEVICES, the following field specifications can be used:

$DEVNAME	Sort on the device name. If no name exists in the circuit, and no name has been assigned using ASSIGNNAMES, then the item will be placed at the end of the list. If the device name contains a numeric portion (e.g. U12), then the numeric part of the name is sorted on its integer value, instead of its character value. E.g. the name U12 will appear before U110, instead of after, as it would in a purely lexical sort.
$DEVHIERNAME	Sort on the device hierarchical name, i.e. a name generated by prefixing the device name with the names of all parent devices in the hierarchy. This is normally only used in flattened netlists.

$DEVINSTNAME	Sort on the contents of the InstName field. This is normally only used in flattened netlists.
$POSX	Sort on the horizontal position of the top left corner of the device symbol on the diagram.
$POSY	Sort on the vertical position of the top left corner of the device symbol on the diagram.
$TYPENAME	Sort on the device's type name, i.e. as it appears in the Parts palette.
$ORIENT	Sort on the device orientation.
$PAGE	Sort on the page number that the device appears on.
$NUMPINS	Sort on the number of device pins.
$DEPTH	Sort on the contents of the Depth attribute field. NOTE: This field will only be valid if the $HIERARCHY $PURE mode is enabled.
&attrField	Sort on the contents of the attribute field named attrField in each device. If that field is empty or does not exist in a given device, then that device will be placed at the end of the list.

If "objectType" is $SIGNALS, the following field specifications can be used:

$SIGNAME	Sort on the signal name. If no name exists in the circuit, and no name has been assigned using ASSIGNNAMES, then the name "unnamed" will be assumed. See the note under "DEVNAME" above regarding sorting numeric portions of the name.
$SIGHIERNAME	Sort on the signal hierarchical name, i.e. the name of the signal prefixed by the names of all parent devices in the hierarchy. This is normally only used in flattened netlists.
$NUMPINS	Sort on the number of device pins attached to this signal.

`$SIGSOURCE`	Sort on the order of appearance of this signal in a SIGSOURCE command. If the signal does not appear in a SIGSOURCE command (as most won't), it will appear after the SIGSOURCE signals. This is used to place all power and ground nets at the front of a netlist.
`&attrField`	Sort on the contents of the attribute field named "attrField" in each signal. If that field is empty or does not exist in a given signal, then that signal will be placed at the end of the list.

The combination options COMBSIGS and COMBDEVS instruct the report generator to combine items with the same sort value into a single line in the listing. This can be used to produce a variety of useful listing formats, for example listings of all devices having a certain type name.

IMPORTANT: The "COMB" options will have no effect if no sort has been specified since they only operate on sorted values.

See the examples at the end of this manual for more information on how to use the sorting capability.

$COMBSIGSON/$COMBSIGSOFF

When ON causes signals with the same sort value to be combined into a single net in the netlist. Default is OFF.

NOTE: COMBSIGSON should not be used in most normal netlisting applications. Any like-named signals have already been logically connected by DesignWorks and will be listed in one entry even if COMBSIGS is OFF. COMBSIGSON will have the additional effect of merging like-named signals, even if they appeared on different pages of the schematic.

$COMBDEVSON/$COMBDEVSOFF

When ON causes devices with the same sort value to be merged into a single entry in the component list. Default is OFF.

The FIND Command

The FIND command provides the ability to select subsets of the circuit data for reporting purposes. This can be used to produce the following types of reports:

- listings containing only objects that are selected on the circuit diagram.
- netlist formats that require different formats for different types of devices or signals. For example, some formats require a list of the I/O ports of a circuit separately from other signals.
- error checking listings. For example, all devices missing a certain attribute field or all signals with less than two pin connections.

The FIND command has the following form:

```
$FIND objectType field1 field2 ...
```

"objectType" must be either "$DEVICES" or "$SIGNALS".

"field1", "field2", etc. are identifiers indicating which fields to find on. After the FIND command is complete, only objects matching all the field specifications will be left in the list. If no field items are specified, all object will be found. This can be used to clear a previous FIND.

Once a FIND has been done, it remains in effect for all subsequent listings on that object type, or until another FIND is done.

IMPORTANT NOTES:

1) If a FIND command is used, it should appear before any other command that refers to the signal or devices lists, such as $DEVICES, $SIGNALS, $SORT, $CHECK and $ASSIGNNAMES. If this is not done, then these other commands will have operated on objects that are not included in the final list, which may give confused results.

2) A FIND command always creates a new list of objects from the design, regardless of previous FIND commands.

The following field options are valid for devices or signals:

`$NOT`	This inverts the meaning of all field specifications following on this FIND command. I.e. find all objects that do not match the given specification.
`$SELECTED`	This selects only objects that are selected (i.e. highlighted) in the circuit diagram.
`&attrField(pattern)`	Select on the contents of the attribute field named attrField in each signal or device. If the pattern is not specified, the object will be found if the given field exists, regardless of contents (even if empty). If the pattern is specified, then the contents of the field must match the pattern exactly. See below for information on patterns.

If "objectType" is $DEVICES, the following field specifications can be used:

`$DEVNAME(pattern)`	Select on the device name. An optional pattern is specified in parentheses after the keyword (see below for more information on name patterns). If the pattern is not specified, then all named devices are selected and all unnamed devices are excluded. If the pattern is specified then the name must match the pattern exactly to be selected.
`$TYPENAME(pattern)`	Select on the name of the device's type, i.e. the name that appears in the Parts palette. If the pattern is not specified, this item will be ignored (i.e. all devices will be selected).

If "objectType" is $SIGNALS, the following field specifications can be used:

`$SIGNAME(pattern)`	Select on the signal name. An optional pattern is specified in parentheses after the keyword (see below for more information on string patterns). If the pattern is not specified, then all named signals are selected and all unnamed signals are excluded. If the pattern is specified then the name must match the pattern exactly to be selected.
`$NUMPINS(min)`	Select signals only with greater than or equal to the number of pins specified.
`$SIGSOURCE`	Select only signals that appear in a SIGSOURCE command.

String Patterns

The SIGNAME, DEVNAME, TYPENAME and attribute field specifications allow an optional pattern to be given in a manner similar to the FIND command in DesignWorks. All objects with a field value matching the pattern will be selected. A number of "wildcard" characters can be used to specify ranges of field values, as follows:

- "*" matches any string of zero or more characters. For example "D*" will match "D", "D0", "D12", "DogGone", etc. "*" by itself matches anything, although the SIGNAME and DEVNAME specifications will not select objects that are unnamed.
- "#" matches any string of 1 or more digits. E.g. "XA#" will match "XA0", "XA123", etc.
- "?" matches any single character. For example, "CTRL?" will match "CTRLX", "CTRL/", etc.

The wildcard characters can be used in combination, as in "DATA#??" which will match "DATA123", "DATA1SQ", etc.

The CHECK Command

The CHECK command is used to detect missing or incorrect data in a circuit file before a report is generated. It has two distinct functions:

- It can be used in combination with the FIND command to locate any devices or signals meeting some error condition (e.g. missing an attribute field). It causes a message to be displayed on the screen if

there were any objects of the given type selected by the last FIND command. Thus, errors are detected by creating a FIND command that selects only objects with the problem in question, then following this with a CHECK command to display the error message.

- It can be used to check that the hierarchy mode of the current design is appropriate for use with this form file. In this form, the $CHECK command is followed by one or more keywords indicating the modes that are incorrect. If the mode of the current design is any of these, the message box is displayed.

IMPORTANT: In a correct circuit, the FIND command used for checking purposes will result in no objects being selected for listing. Therefore you must follow the CHECK command with another FIND otherwise no report will be generated.

The format of the CHECK command is one of the following:

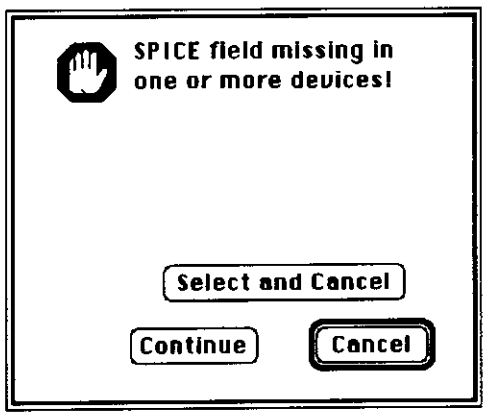
```
$CHECK(message) objectType
$CHECK(message) modes
```

- "objectType" is either "$DEVICES" or "$SIGNALS".
- "modes" is one or more of "$PURE", "$PHYSICAL", or "$FLAT".
- "message" is an optional text message to be displayed if an object of the given type is found on the list. If the message is not specified, a standard message will be displayed.

For example, the following sequence will display a message if the attribute field "SPICE" is missing from any devices:

```
$FIND $DEVICES $NOT &Spice
$CHECK(SPICE field missing in one or more devices!) $DEVICES
$FIND $DEVICES { Re-find all devices }
```

This will cause the following box to be displayed if the error condition is detected:



The "Select and Cancel" option gives the user the opportunity to have the offending devices or signals highlighted on the circuit diagram.

Listing Commands

The $DEVICES and $SIGNALS commands are the heart of the form file in that they cause the actual listing to be generated. The format of the listing is determined by the format specification that follows the command on the line and by the various options that have been discussed above.

Note the following general rules regarding these commands:

- in a basic listing, one line is generated for each item in the list, but this will be affected by the following:
  - if the COMBDEVS or COMBSIGS flags are on, multiple items with the same sort value will appear on one line.
  - the $NEWLINE format command can force multiple lines to be generated for each item.
  - limits on line length imposed by the $LINEWIDTH and $MAXITEMSPERLINE commands may cause multiple lines to be generated.
- if the COMBDEVS or COMBSIGS flags are used to merge multiple items into one line, some of the format commands may be ambiguous. For example, if a device list is sorted by type name and COMBDEVS is ON, each line may refer to multiple devices. In this case, a format command such as $DEVNAME will output the names of all devices covered by each line. This can be used to create a bill of materials

which lists each device type, followed by the names of all instances of that type.

• the format of the listing for each device or signal is completely determined by the text and commands which follow the $DEVICES or $SIGNALS keyword. There is no default format, so if no line format is specified, a sequence of blank lines will be written to the file.

• DesignWorks "pseudo-devices" such as page connectors, signal connectors and bus breakouts are never included in any listing.

• any characters appearing after $DEVICES or $SIGNALS that is not part of a format sub-command will be placed verbatim in each line of the listing. For example, blanks, tabs or commas can be used to format each line.

Listing by Signal

`$SIGNALS lineFormat`

The SIGNALS command creates a listing of the signals in the circuit, along with any related data such as signal names, signal attribute data and the names of the devices and pins that a signal is attached to. The format of each line is completely specified by the data following the $SIGNALS keyword on the line.

Sub-Commands to be used after SIGNALS

`$SIGNAME` - writes out the name of the signal.

`$SIGSEQ(n)` - writes the sort sequence number of the signal. These numbers are assigned in order, starting with the number "n", when the signals are sorted. If "n" is not specified, then numbering will start with zero. NOTE: If no SORT has been done, all signals will have sequence number "n".

`$SIGTOKEN` - writes the signal token number that was assigned by DesignWorks when the signal was first created. This number will be unchanged for the life of the signal in this circuit.

`$BUSNAME(`*`separator`*`)` - if the signal being listed is contained in a bus, then the name of the bus is inserted, followed by the separator text. If the signal is not contained in a bus, then nothing is inserted.

`$COUNT` - outputs a count of the signals with the same sort value that have been combined on this line. If COMBSIGSOFF has been set, this will always be 1.

$PINS - list all of the pins attached to the signal by their device names and pin numbers. The format of each pin item is determined by the $SIGPINFORMAT command.

$SINGLE - tells the report generator to write only a single value for all items following on the line, regardless of how many items were combined with the same sort value.

$PAGE - lists all circuit pages that this signal appears on.

$COL(N) - If the current position in a line is less than N spaces form the left hand end then blanks are inserted until the Nth column is reached. For instance $COL(20) will force any output to be indented to the 20th column of each line.

$NEWLINE - Writes a carriage return character into the file.

&attr - Writes the contents of an attribute field of the given name. For example a field called WIDTH could be defined for signals using the Define Attribute Fields command. The Report Generator will extract information from these fields by using the command &WIDTH which could be used to print out the width of each trace to be passed to a PCB layout package.

Examples:

```
$SIGNALS$SIGNAME $PINS
```

This line will produce a basic netlist, with each line consisting of the signal name, a blank and a list of the device pins (device name and pin number) attached to this signal, as in:

```
CLOCK U1-12 U6-19 U7-12
```

The default pin format is shown above. This can be changed by the SIGPINFORMAT command, as follows:

```
$SIGPINFORMAT $PINNUM$DEVNAME
$SIGNALS#$SIGSEQ $PINS
```

will produce a netlist of the following form:

```
#00001 12U1 19U6 12U7
```

See the more detailed examples later in this manual.

Listing by Device

```
$DEVICES lineFormat
```

The $DEVICES command generates a listing of the devices and their related data, such as the device name, the device type and any device attributes. "lineFormat" consists of any combination of random text and the following sub-commands.

$DEVNAME - the name of the device.

$DEVSEQ(n) - the sequence number of the current device, i.e. an integer assigned to the device, starting at "n" for the first device in the sorted sequence and incremented for each subsequent one. This number is not permanently associated with the device but it strictly its sequence number in the current sort. The "(n)" part is optional. If it is omitted the sequence starts at zero. NOTE: If no sort has been done, all devices will have sequence number "n".

$DEVTOKEN - writes the device token number that was assigned by DesignWorks when the device was first created. This number will be unchanged for the life of the device in this circuit.

$TYPENAME - the device's type name. I.E. the name used to refer to this type in the library.

$COUNT - the number of devices being merged to form this line, i.e. the number having the same sort value.

$POSX - the X coordinate of the top left point on the device symbol in 1/1000 in.

$POSY - the Y coordinate of the top left point on the device symbol in 1/1000 in.

$SINGLE - tells the report generator to write only a single value for all items following on the line, regardless of how many items were combined with the same sort value.

$ORIENT - the orientation of the device symbol. One of the following direction codes will be output.

Output String	Direction
E1	East
E2	East flipped
N1	North
N2	North flipped

W1	West
W2	West flipped
S1	South
S2	South flipped

`$NUMPINS` - the number of pins on the device.

`$NUMINPS` - the number of inputs on the device. This number is valid only on primitive devices.

`$NUMOUTS` - the number of outputs (i.e. anything other than an input) on the device. This number is valid only on primitive devices.

`$PAGE` - the circuit page that this device appears on.

`$GRID` - the page reference grid position.

`$PINS` - a list of the device's pins. The format of each item is determined by the $DEVPINFORMAT command. If a DEVPINFORMAT command is not specified, the default pin format shows only the name of the attached signal. The order in which the pins are written out is determined by the following rules (in descending order of priority):

- if a $DEVPINSEQUENCE command has appeared previously in the form file, and the specified attribute field exists in the device, then the pins are written in the specified order.
- if a $PINNUM appears in the DEVPINFORMAT, the pins are sorted in order of the this pin number.
- if neither of the above applies, then the pins are written in the order they were created, i.e. the order they appear in the DevEditor's pin list.

`$COL(N)` - If the current position in a line is less than N spaces form the left hand end then blanks are inserted until the Nth column is reached. For instance $COL(20) will force any output to be indented to the 20th column of each line.

`$NEWLINE` - Writes a new line character into the file. This allows a device entry to occupy several lines in the output report.

`&attr` - Inserts the value of the attribute field specified.

Report searches in a number of places to find device attribute fields, in the following order:

- the attributes for this device
- the attributes for all devices with the same sort value (if COMBDEVSON has been set)
- the attributes for the device type (i.e. as set in the library using the DevEditor)

One result of the above search order is that default attribute values for devices can be specified in the library using the DevEditor. These can then be overridden if necessary by specifying the same field in each instance.

Other Listing Commands

The following items can appear in the body of the form to provide counts of the objects listed in the report. They are only valid after a listing has been generated with the given object type since the count is made as the listing is generated.

$SIGCOUNT	The number of items in the last signals listing. If COMBSIGSON has been set, this will mean in effect the number of different sort values in the list.
$PINCOUNT	After a netlist, this will be a count of the total number of device pins listed.
$DEVCOUNT	The number of items in the last devices listing. If COMBDEVSON has been set, this will mean in effect the number of different sort values in the list.
$CHANGECOUNT	The number of changes (i.e. editing operations) performed on the circuit since it was created. This number can be used to check for matching versions of a circuit file.
$NEWPAGE	The NEWPAGE command causes a page feed to occur. The exact data written out by NEWPAGE depends upon the LINESUSED and PAGELENGTH settings.
$DATE	Writes out the current day, month and year.
$TIME	Writes out the current system time.
$DESIGNNAME	Writes out the name of the circuit file being used as input for the report.

&attr	If an attribute variable appears anywhere in the body of the form file (i.e. not after a $SIGNALS or $DEVICES keyword) then the program attempts to match the variable name with a design attribute field. If a match is found, the value of the field is substituted. This is intended to accommodate global items such as titles, circuit parameters, routing instructions or other information that is not related to any one device or signal.

Hierarchical Net Listing

The Report Generator has the ability to produce a hierarchical netlist of a circuit, i.e. a list including the internals of hierarchical blocks. The listing can proceed into nested blocks to any desired depth to provide the required detail. For the following discussions, this facility is referred to as "hierarchical netlisting". Without hierarchical netlisting enabled, only objects in the current circuit level (i.e. that displayed in the topmost window) will appear in a netlist.

See the hierarchical netlist example in the Report Form Examples section of this chapter for more information.

Types of Hierarchical Netlists

Two types of netlists can be produced from a hierarchical design:

- Flattened Netlist: A flattened netlist is generated by substituting each hierarchical block symbol with its internal circuit. All information about the hierarchical blocks themselves, nesting levels, etc. is lost. The resulting output appears to have been generated from a flat design. This format is most commonly used for printed circuit board netlists.
- Pure Hierarchical Netlist: A Pure Hierarchical Netlist consists of a netlist of the master circuit for the design, and then separate netlists of the internals of each type of hierarchical block used. Each internal circuit can be written into a separate file, or all netlists may be concatenated in a single file in either top-down or bottom-up order. This format is commonly used for simulators (e.g. SPICE) and FPGA tools.

IMPORTANT NOTE: Since a Pure Hierarchical Netlist contains only a single copy of the definition of each internal circuit, it cannot contain any instance data.

See Chapter VII - Hierarchical Design for more information on instance vs. definition data.

The $HIERARCHY Command

The $HIERARCHY command in a form file sets the type of netlist output to be generated.

NOTE: The hierarchy mode of the netlist does not have to match that of the design. For example, you can generate a PURE netlist from a PHYSICAL mode design. The following table indicates which design modes are appropriate for the various netlist types.

Report Mode	Design Mode	Description
$CIRCUIT	FLAT PHYSICAL PURE	Generate a normal netlist for the single current circuit only. This is the default mode.
$TOPCIRCUIT	FLAT PHYSICAL PURE	Generate a normal netlist for the master circuit of the current design only.
$FLAT	FLAT[1] PHYSICAL	Generate a flattened netlist for the entire design.
$FLATDOWN	FLAT[1] PHYSICAL	Generate a flattened netlist for the current circuit and all nested blocks.
$PURE	FLAT[2] PHYSICAL[3] PURE	Generate a pure netlist for the entire design. In particular, this causes a $DEVICES listing to list each unique type of device only once.
$PUREDOWN	FLAT[2] PHYSICAL[3] PURE	Generate a pure netlist for the current circuit and all nested blocks.

Notes:

1) A flattened netlist will normally make reference to instance attributes in the design. Instance attributes are not normally used in a Flat mode design, so this combination is not recommended.

2) PURE netlist mode can be used with a FLAT design but no internal circuits will be listed.

3) In PURE netlist mode, the internal circuit for each block type is listed only once, regardless of how many times it is used in the design. Therefore, instance data is not normally included in a PURE netlist.

Restricting Reporting of Internal Circuits

The listing of internal circuits of individual blocks can be controlled using the Report Options button in the Get Info command for a device. The option "Report this device" will cause this device or block to appear in the netlist as if it has no internal circuit. The option "Report sub-circuit" will cause the internal circuit of this device will appear in the netlist. I.e. in a PURE netlist, the internal circuit will be defined, in a FLATTENED netlist, the internal circuit will be substituted and this device will not appear at all.

Listing Format for Internal Circuits

For flattened netlists, the output format is usually the same as would be generated from a flat design, except that hierarchical or instance names (see next section) are sometimes substituted for definition names.

For Pure Hierarchical Netlists, formatting is more complex since the format of the master circuit is usually different from the internal circuit definitions. For this reason, the Report tool treats these two types of circuits as two separate format definitions.

The general outline of a Pure Hierarchical Netlist format is as follows:

```
{
  $PURE mode lists each device type once
  so the next $DEVICES listing will only
  list unique type definitions.
}
$HIERARCHY $PURE
{
  The DEFINECIRCUIT section gives the listing
  format for the internal circuits
}
$DEFINECIRCUIT
{ Listing commands for internal circuit }
$END
{
  Use FIND command to locate all devices that
  have a non-empty Depth attribute, i.e.
  they have an internal circuit.  The Depth
  attribute field contains an integer indicating
  the number of circuit levels below this device
}
$FIND $DEVICES &Depth
{
  Sort by the Depth field to list from the
  bottom up.  The $DESCENDING option could
  be used to sort from the top down.
}
$SORT $DEVICES &Depth
{
  The following $DEVICES listing will list
  each unique type of device only once because
  we are in $PURE mode.  $INTERNAL causes the
  internal circuit to be output for each device
  in the format defined in $DEFINECIRCUIT.
  Other text or commands could be included in
  the $DEVICES line to output a circuit block
  header or terminator.
}
$DEVICES$INTERNAL
{
  Now switch to TOPCIRCUIT mode and do a
  separate listing for the master circuit
}
$HIERARCHY $TOPCIRCUIT
{ Commands to list master circuit }
```

Depth Ordering in Pure Netlists

The Depth attribute field is used to select device types that have internal circuits and to sort by depth for top-down or bottom-up listings.

> NOTE: The Depth field is not maintained by the Schematic module during normal editing operations. The Depth value is calculated by the Report tool only when a PURE netlist is generated.

The Depth field will be empty for devices with no internal circuit or those with the "Report This Device" option selected in the device Report Options. For devices with an accessible internal circuit, the Depth field will contain an integer indicating the number circuit levels below the device. E.g. if the device's internal circuit contains only bottom-level devices, the value will be 1.

The Depth field can therefore be used with the $FIND command to perform separate listing operations on devices with or without internal circuits. For example:

```
$FIND $DEVICES &Depth
```

will select all devices having an internal circuit, whereas:

```
$FIND $DEVICES $NOT &Depth
```

will select devices having no internal circuit. Similarly, the $SORT command can be used to determine whether internal circuits are defined from the lowest level up or vice-versa.

```
$SORT $DEVICES &Depth
```

will cause the lowest level devices to be listed first, i.e. bottom up. This is the preferred order for most formats because it ensures that each type of internal circuit is defined before it is referred to in any other circuit.

Instance vs. Definition vs. Hierarchical Names

Devices and signals have three different types of "names" that are significant in producing netlists from hierarchical designs. These are summarized in the following table.

Report Keyword	Attribute Field	Description
$DEVNAME $SIGNAME	Name	Normally used in Flat mode designs. Not guaranteed to be unique across a hierarchical design.
$DEVINSTNAME $SIGINSTNAME	InstName	Not normally used in Flat mode designs. InstName is used for package assignment in Physical mode designs. Should be unique across design.
$DEVHIERNAME $SIGHIERNAME	(temporary generated by Report)	A generated name consisting of the device or signal Name field prefixed by the names of all containing hierarchical blocks.

Note the following rules when selecting which type of name to use in a report format:

- $DEVNAME/$SIGNAME should not be used in a flattened netlist unless you intend to manually guarantee that it is unique across the entire design and each type of hierarchical block is only used once.
- In Physical mode designs, $DEVINSTNAME should be used as the device "name" in netlists intended for PCB-layout use, since it normally contains the package assignment.
- In flattened netlists, $SIGHIERNAME should be used for the signal "name" since it ensures a unique name that is easily located in the schematic. $SIGINSTNAME can be used but must be assigned manually to each signal.

Power and Ground Connections in Hierarchy

Power and ground symbols (i.e. Signal Connector devices) do not make an immediate logical connection across hierarchy levels. For this reason, power and ground connections in different circuit levels will be considered separate nets in a flattened netlist, unless you do one of the following:

- Specify the name of each power and ground net in a $SIGSOURCE statement in the form file. This instructs the Report tool to look for those named nets and integrate them into a single entity.
- Create individual ports for the power and ground connections on each hierarchical block symbol and treat them as regular nets.

Report Command Summary

This section lists the keywords available in the various parts of a form file.

Definition Commands

```
$ALIGNCOLSON/$ALIGNCOLSOFF
$ATTRNUM(number) &attrField
$AUTONUMBER(number of pins)
$BLANKREPLACE(string)
$CLOSEREPORT
$COMBPINSON/$COMBPINSOFF
$CONTEND(string)
$CONTREPEATON/$CONTREPEATOFF
$CONTSTART(string)
$CREATEREPORT(reportName) $PROMPT $CREATOR(type)
$DEFINECIRCUIT/$END
$DEVPINSEQUENCE &attrField
$HIERARCHY
   $FLATDOWN $FLAT $CIRCUIT $TOPCIRCUIT
   $PUREDOWN $PURE
$HIERNAMESEPARATOR
$ITEMSEPARATOR(string)
$LINESUSED(n)
$LINEWIDTH(n)
$MAXITEMSPERLINE(n)
$NOTES/$END
$ONEPINSON/$ONEPINSOFF
$PAGELENGTH(n)
$REPEATITEMON/$REPEATITEMOFF
$SIGSOURCE(signalName) &fieldName
$SPACE(number)
$UNCONNPINSON/$UNCONNPINSOFF
$UNNAMEDDEVS(string)
$UNNAMEDSIGS(string)
```

Item Format Commands

```
$DEVPINFORMAT/$SIGPINFORMAT

   $CHILDSIGNAME
   $DEVHIERNAME
   $DEVINSTNAME
   $DEVNAME
   $DEVSEQ
   $DEVTOKEN
   $GRID
   $NEWLINE
   $PAGE
   $PINDIR
   $PINNAME
   $PINNUM
   $PINSEQ
   $PINTYPE
   $POSX
   $POSY
   $SIGHIERNAME
   $SIGINSTNAME
   $SIGNAME
   $SIGNUM
   $SIGSEQ
   $SIGTOKEN
   $SYSPIN

$PINTYPEFORMAT string1 string2 ... (17 items)
```

Page Header Format

```
$PAGE
$DATE
$TIME
$DESIGNNAME
$ENDHEADER
&attr
```

Selection and Sorting Commands

```
$ASSIGNNAMES type $FORMAT(pattern) &field
        $TYPENAME

$SORT objectType field1 field2 ...

$COMBDEVSON/$COMBDEVSOFF
$COMBSIGSON/$COMBSIGSOFF

$FIND objectType field1 field2 ...

$CHECK(message) objectType
```

Listing Commands

```
$SIGNALS format

   $BUSNAME(separator)
   $COL(N)
   $COUNT
   $NEWLINE
   $PAGE
   $PINCOUNT
   $PINS
   $SIGHIERNAME
   $SIGINSTNAME
   $SIGNAME
   $SIGSEQ(n)
   $SIGTOKEN
   $SINGLE
   &attr
```

```
$DEVICES format

   $COL(N)
   $COUNT
   $DEVHIERNAME
   $DEVINSTNAME
   $DEVNAME
   $DEVSEQ(n)
   $DEVTOKEN
   $GRID
   $INTERNAL
   $NEWLINE
   $NUMINPS
   $NUMOUTS
   $NUMPINS
   $ORIENT
   $PAGE
   $PINCOUNT(&seqField)
   $PINS
   $POSX
   $POSY
   $PRIMNAME
   $SINGLE
   $TYPENAME
   &attr

$CHANGECOUNT
$DATE
$DESIGNNAME
$DEVCOUNT
$NEWPAGE
$PINCOUNT
$SIGCOUNT
$TIME
&attr
```

Chapter XII - Browser Tool

Introduction

The Browser is a tool used to view, edit and navigate schematic data in spreadsheet-style format. Rows of the spreadsheet display objects in the schematic, either devices, signals, pins or nets. A cell of the spreadsheet displays an attribute value for the object in that row and the attribute field name in that column.

Installation

The Browser is a code module that works with DesignWorks using the MEDA interface. To locate the module when DesignWorks starts up, it must be in the same folder with the DesignWorks program itself, or in the folder specified in the TOOLFOLDER statement in the DesignWorks setup file, normally the Tools folder.

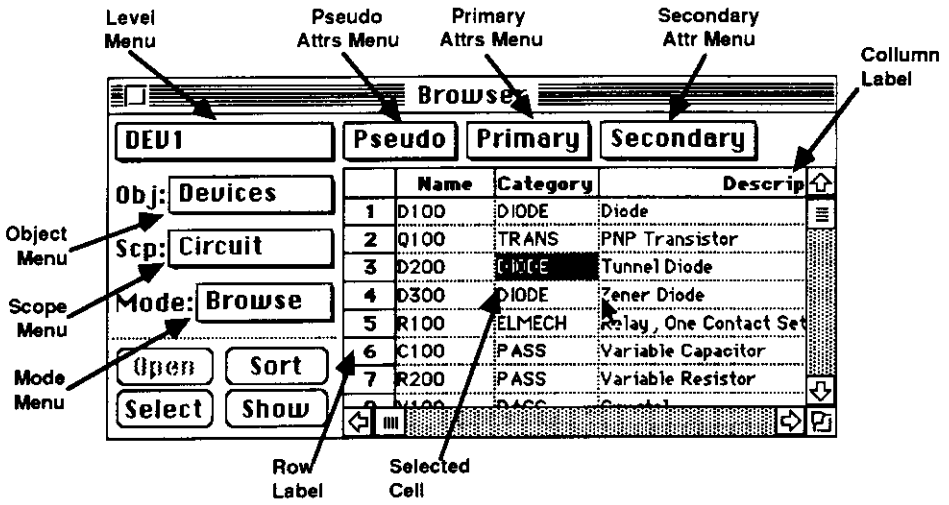
Memory Usage

The Browser tool occupies a small amount of memory space, about 82K. It may however require a much larger amount of memory depending on the amount of information displayed in the spreadsheet. If you are operating on a system with a minimal amount of memory, it may be desirable to keep the Browser tool in another folder where it will not be found and loaded by DesignWorks. When needed, it can be moved temporarily into the DesignWorks folder.

Starting the Browser

To invoke the Browser select it from the Tools menu. This opens a spreadsheet window. By default the Browser will initially display all devices in the current circuit. If any objects are selected in the schematic, the Browser will display only selected devices or signals. Selected devices have priority over selected signals.

The Browser Window



The following table summarizes the features available in the Browser window.

Level Menu	A popup menu that displays the name of the current circuit level in a design being viewed. When selected, the popup menu shows the circuit hierarchy upwards from the current circuit.
Pseudo Attr Menu	When selected the popup menu shows attributes of the object that are not true attributes, e.g. Grid Position.
Primary Attr Menu	When selected the popup menu shows attribute names that may be displayed for the selected object type and that are distinguished as primary in the attribute definition dialog.
Secondary Attr Menu	When selected the popup menu shows the attribute names that may be displayed for the selected object type and that have not been distinguished as primary in the attribute definition dialog.

Column Label	Except in Net mode, this displays the name of the attribute data shown in the column. In Net mode the columns are simply numbered.
Selected Cell	The selected cells are operated on by the Open, Select, and Show buttons, as well as the command popup menu.
Row Label	Generally it just contains a row index, but in Net mode it contains the name of the net associated with the row.
Mode Menu	A popup menu which when not selected displays the current mode. When selected it allows the user to choose between Browse and Edit mode.
Scope Menu	A popup menu which when not selected displays where the Browser will look for information. When selected it allows a user to choose between Design, Circuit, Here Down, and Page.
Object Menu	A popup menu which when not selected displays what type of object the Browser is displaying. When selected it allows a user to choose between Devices, Signals, Pins, Nets, Selected Devices, Selected Signals, Selected Pins, and Selected Nets.

Updating the Browser Window

The Browser will update any time a change occurs in the schematic that effects the data displayed in the spreadsheet, or when its scope is set to Circuit and the current circuit changes.

NOTE: The update is delayed until the mouse is in the Browser's window and the window is at the front.

Browser Procedures

Selecting Which Objects are Displayed

The object menu popup allows the user to see and change the type of object being displayed. There are eight different types of objects that may be displayed Devices, Signals, Pins, Nets, and selected Devices, Signals, Pins, and Nets.

If the selected form of an object is chosen to be displayed, e.g. Selected Devices, then only objects of the same type, i.e. Devices, which are selected in the schematic will be displayed. Another example, if Selected Signals is chosen in the object menu popup then only signals that are selected in the schematic will be listed in the spreadsheet.

Clicking on the object menu popup or making a selection from it will cause a new search to be performed to find the requested objects. The attribute fields displayed the last time this object type was selected will again be displayed.

Determining Where to Search for Objects

When looking for objects, the Browser will limit its search scope by using the selection made in the scope menu popup. There are four different scopes:

Design. In this scope the entire design hierarchy is search for objects of the correct type. This scope has a side-effect. Some of the objects attribute values displayed may not be editable because of definition/instance conflict resolution. The uneditable objects and/or fields will be displayed with a gray background.

Circuit. In this scope the search is limited to the pages of the circuit with the most active window.

Here Down. In this scope the search for objects is in the circuit with the active most window and in all circuits below it in the hierarchy. Some of the objects and/or fields may not be editable for the same reasons as explained in Design scope.

Page. This scope only looks for objects in the circuit with the most active window and in addition only looks at the page associated with that window.

Displaying Attributes

For all objects other then Nets and Selected Nets, attributes may be displayed. Attributes are divided into three groups:

Pseudo Attributes. These are characteristics about the object that the schematic knows but are not true attributes. These attributes may not be edited.

Primary Attributes. These are attribute that have been selected in the Define Attribute Fields box in the Schematic Tool and marked as being primary. This is done to allow easier access to the more common attributes.

Secondary Attributes. These are all the attributes that are not marked as primary (see above).

The attribute field names displayed in each popup menu will change as the type of object selected for display is changed. Only attributes that apply to a given object type are displayed. For example, the attribute field "Part" is associated with Devices but will not be associated with Signals.

In addition, some attribute names in the Primary and Secondary attribute menus will be underlined. An attribute name that is underlined indicates that the attribute value is stored with the instance of the object and not with its definition.

See Chapter VI - Attributes for more information on defining attribute fields.

If an attribute name is selected in one of the popup menus then it will be added as a new column at the end of the spreadsheet. In addition, the attribute name in the menu will have a check mark (√) beside it. Selecting a checked attribute name removes the associated attribute from the spreadsheet.

Changing Attribute Values

Selecting & Editing Cells

There are two modes of operation:

Browse mode allows the user to look at the objects and their attributes in the spreadsheet but not to edit the text of a cell.

Edit mode allows all the same operations as Browse, but in addition allows cells to be edited.

A cell may be selected by clicking in its bounds. A range of cells may be selected by clicking in another cell with the shift key down. The selected cells will be inverted. If the user is in edit mode and the cell is editable (no gray background) then the last cell clicked in will have a flashing edit cursor in it.

Modifying the value of a cell will cause the corresponding attribute value of the object in the schematic to change. The change happens **immediately** and there is **no undo**.

Using the Command Popup Menu

Pressing the mouse button with the ⌘ key down, while over the spreadsheet, will cause a popup menu to appear for all objects except Nets. This popup allows cells, in Browse mode, or the text in cells, in Edit mode, to be Cut, Copied, Pasted, and Cleared. There are four additional menu items:

Fill Down	The Fill Down command takes the selection rectangle and copies the values in the cells at the top of the selection to all the selected cells below them.
Fill Right	The Fill Right command takes the selection rectangle and copies the values in the cells at the left of the selection to all the selected cells to the right of them.
Make Visible	The Make Visible command takes the object attributes affected by the selection rectangle and makes them visible in the schematic.
Make Invisible	The Make Invisible command takes the object attributes affected by the selection rectangle and makes them invisible in the schematic.

Disallowed Editing Changes

Any editing change that may cause a structural change to the connectivity of the schematic will be disallowed and aborted by the Browser. There are only a few special cases that will cause this to happen. For example:

- Renaming a visible signal to the same name as another visible signal on the same page would cause the two signals to become one signal.
- Making a signal visible which has the same name as an already visible signal with the same name would cause the two signals to become one signal.

Using the Buttons

Select Button

The Select button will be enabled if there are objects selected in the spreadsheet. Pressing the button will cause the objects to be selected in the schematic but will not cause the schematic to be repositioned.

Note: If the object type being viewed is Selected Device, Selected Signal, Selected Pin, or Selected Net then this operation will cause the spreadsheet to update to display the new selected objects.

Show Button

The Show button will only be enabled if there is a single object selected in the spreadsheet. Pressing the button will cause the schematic to be repositioned so that the object is visible. In addition, the object will also be selected and all other objects in the same circuit will be deselected. Note this will have the same side effect as above if the object type is Selected Device, Selected Signal, Selected Pin, or Selected Net.

The Show button may sometimes not be able to show the object. When this happens a dialog will be displayed. The reason for this is generally that the object was not in an open window.

Open Button

The Open button will be enabled if there is a single object selected in the spreadsheet that has a subcircuit either directly or indirectly. Objects that may be opened are Devices and Pins (both for Pin objects and Pin connections for Net objects). Pressing the button will cause the subcircuit to be opened.

Sometimes it is not possible to open the subcircuit. When this happens a dialog will be displayed. The general reason for not being able to open the subcircuit is that a copy of the circuit is already open for another instance of the part.

Sort Button

Pressing the Sort button causes the rows of the spread sheet to be sorted into increasing order.

If the object type is not Net or Selected Net then the following comparison mechanism is used when comparing two rows:

- The values of the first column are compared. If the values are not the same then this is sufficient to determine the sort order.
- If the values were the same then the next column is used to determine the sorting order. This is repeated as often as needed or until both rows are determined to have the same sort value.

If the object type is Net or Selected Net, then the rows are not sorted by the contents of the spreadsheet, but are instead sorted by the net's name that is in the row label.

Adjusting the Spreadsheet

Sizing Rows & Columns

Rows and columns may be resized by moving the cursor into the row or column heading area and positioning it over the horizontal (row) or vertical (column) dividing line. The cursor when correctly positioned it will change shape. Press the mouse button and drag the line to size the row or column to its desired size.

Moving Rows & Columns

Rows and columns are reordered by moving the cursor into the row or column heading area and positioning it over the row or column heading to move. The cursor will change shape to indicate that it is correctly positioned. Press the mouse button and drag the label to its new position.

Note: It can be useful to move columns so that the sorting mechanism will base the sort on the new ordering of the columns.

Chapter XIII - Find and ErrorFind Tools

Introduction

Find and ErrorFind are tools used to locate devices, signals or pins in a schematic. Once the objects are found they may be sequentially selected or selected as a group.

Find searches by name or attribute value whereas ErrorFind searches for possible error conditions. After errors are found they may be selected and marked as OK or corrected.

Installation

Find and ErrorFind are code modules that work with DesignWorks using the MEDA interface. To locate the modules when DesignWorks starts up, they must be in the same folder with the DesignWorks program itself, or in the folder specified in the TOOLFOLDER statement in the DesignWorks setup file, usually called Tools.

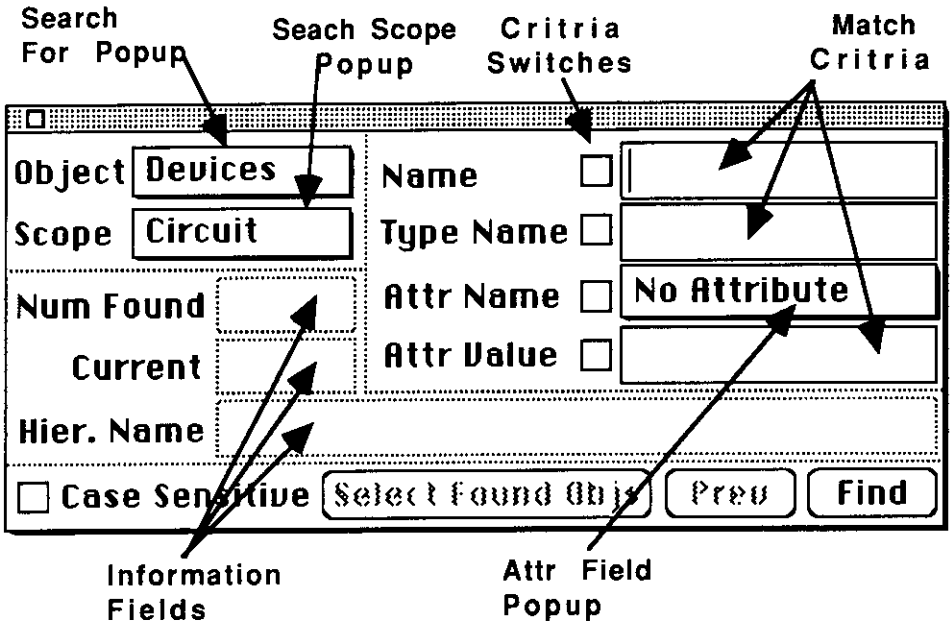
Memory Usage

The Find and ErrorFind tools each occupy a small amount of memory space, about 38K. In addition they require enough memory to record each object found that matched the search criteria. If you are operating on a system with a minimal amount of memory, it may be desirable to keep the Find tool in another folder where it will not be found and loaded by DesignWorks. When needed, it can be moved temporarily into the DesignWorks folder.

Starting Find or ErrorFind

To invoke Find or ErrorFind, select the name in the Tools menu. This opens a find palette window.

The Find Window



Search Scope

Search for Popup A popup menu that displays the name of the type of object to be found. When clicked on, the popup menu allows the user to select one of the following choices Devices, Signals, Pins, Selected Devices, Selected Signals, or Selected Pins.

Search Scope Popup A popup menu that displays the name of the scope to search for objects. When clicked on, the popup menu allows the user to select one of the following choices.

Design	All unlocked circuits in the design are searched.
Circuit	Only the current circuit is searched.
Here/Down	All unlocked circuits from the current circuit and down in the hierarchy are searched.
Page	Only the current page is searched.

Search Criteria

Criteria Switches

These checkboxes indicate if their corresponding criteria will be used when trying to find a match. The check box will be automatically selected if a value is entered in the corresponding field.

Match Criteria

Text entered into these fields is compared to the values in the objects examined to see if they match. The following "wildcard" characters may be used:

*	Any string of zero or more characters
?	Exactly one character
#	Any string of one or more decimal digits

Attr Field Popup

A popup menu that displays the name of the attribute field to be matched. When clicked on, the popup menu allows the user to select one of the attribute fields valid for the type of object being searched for.

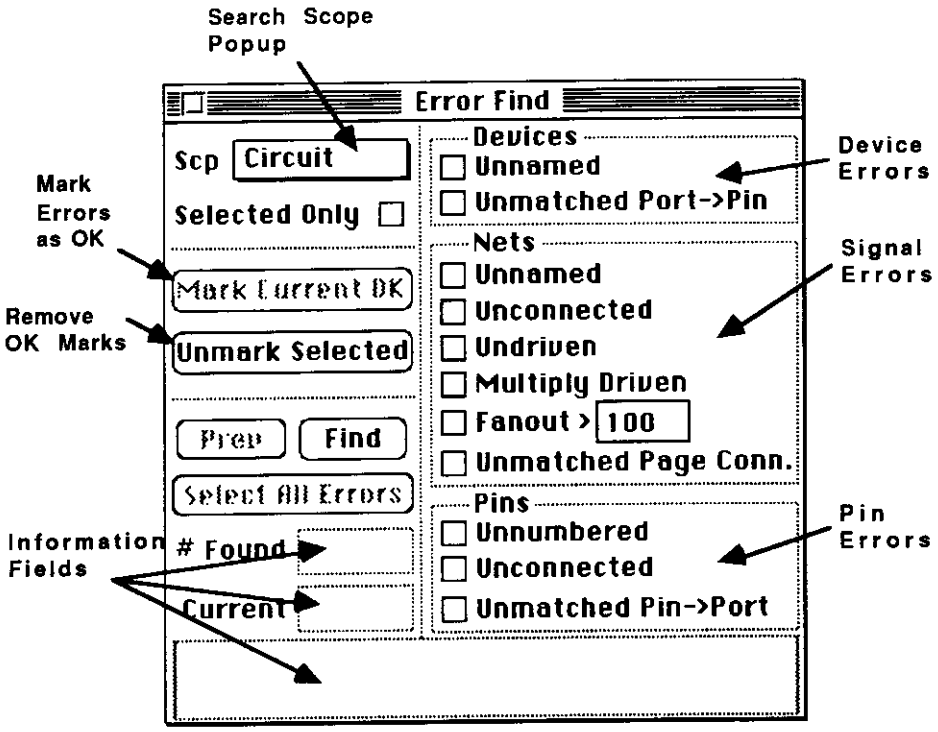
Information Fields

These fields may not be edited by the user but instead display results back to the user. The top field displays the total number of objects found that matched the criteria. The next field indicates which of the found objects is currently selected. The bottom field displays the objects hierarchical name, i.e.. the object's name prefixed by its position in the hierarchy.

Find Controls

There are three buttons. The first "Select Found Objs" selects all of the objects found. The second button "Prev" moves backwards in the list of found objects and selects the object before the current one. The last button "Find" / "Next" has two operations. Initially, when the button is labeled "Find", it is used to initiate the search. Once the objects have been found then its label changes to "Next" and when pressed it advances the selection to the next found object. You can return to the "Find" by changing any of the search criteria or by making a structural change to the circuit.

The ErrorFind Window



Search Scope

Search Scope Popup

A popup menu that displays the scope of the search for errors. When clicked on, the popup menu allows the user to select one of the following choices.

Design	All unlocked circuits in the design are searched.
Circuit	Only the current circuit is searched.
Here/Down	All unlocked circuits from the current circuit and down in the hierarchy are searched.
Page	Only the current page is searched.

Selected Only Option

If the Selected Only box is checked, only objects selected on the schematic will be checked for errors.

Device Errors

Two types of device errors may be searched for:

Unnamed	Any device with an empty Name field.
Unmatched Port to Pin	Port connectors which don't have a corresponding pin on the parent device.

Signal Errors

Six types of errors associated with signals may be located. They are:

Unnamed	The signal's Name field is empty.
Unconnected	The signal has one or zero pins attached to it.
Undriven	No output pins are attached to the signal.
Multiply Driven	Multiple output pins are attached to the signal.
Fanout > Limit	The number of input pins attached to the signal is greater than the specified amount.
Unmatched Page Connector	The signal has exactly one page connector attached to it.

Pin Errors

Three types of errors are associated with pins, they are

Unnumbered	The pin has no pin number.
Unconnected	The pin is unconnected, i.e. no signal line has been attached to it and it is unnamed.
Unmatched Pin to Port	The device has an internal circuit but this pin has not been matched with a port connector.

Ignoring Errors

ErrorFind allows you to indicate that certain errors should be ignored on specific objects. This facility is controlled by the following two buttons.

Mark Current OK	The particular error that caused the object to be found is disabled for the current object, i.e. the object whose name is in the status box.
Unmark Selected	All selected objects in the schematic are "unmarked", i.e. they will be checked for all errors in the future.

Information Fields

These fields may not be edited by the user but instead display results back to the user.

# Found	This field displays the total number of objects found that matched the criteria.
Current	This field indicates which of the found objects is currently selected.
status box	The status box at the bottom of the window displays the object's hierarchical name, i.e.. the object's name prefixed by its position in the hierarchy.

Other Controls

Select Found Objs	Selects all of the objects found.
Prev	Moves backwards in the list of found objects and selects the object before the current one.
Find/Next	This button has two operations. Initially, when the button is labeled "Find", it is used to initiate the search. Once the objects have been found then its label changes to "Next" and when pressed it advances the selection to the next found object. You can return to the "Find" by changing any of the search criteria or by making a structural change to the circuit.

Appendix A - Device Pin Types

Every device pin has a characteristic known as its *pin type*. The pin type is set when the part entry in the library is created and cannot be changed for individual device pins on the schematic.

> *See Chapter X - Device Symbol Editing for information on how to set the pin type while creating a device symbol.*

What Pin Types are Used For

For many general schematic editing purposes, the pin type will be unimportant an can be ignored. However, pin type settings are important in the following cases:

- Pin type settings are crucial if you plan to use the DesignWorks Digital Simulation option. The pin type of each pin determines the direction of signal flow and must be appropriately set for each symbol.
- Pin type information is required in many netlist file formats for FPGA layout and digital simulation.
- Correct pin type settings allow the ErrorFind module to check for fanout and multiple drive situations.
- Other future analysis tools may use this information for timing and loading analysis.

Pin Types Table

The following table lists the function of each of the pin types available in DesignWorks.

Pin Type	Description
IN	Input - this is the default for pins created using DevEditor. This setting is used for all pins on discretes except those with some digital function.
OUT	Output - always enabled.
3STATE	Output - can be disabled (i.e. high-Z)
BIDIR	Bidirectional
OC	Open collector output - i.e. pulls down but not up

BUS	Bus pin - This does not represent a physical signal but is a graphical representation of a group of internal pins, each having its own type.
LOW	Output - always driving low
HIGH	Output - always driving high
LTCHIN	Input to a transparent latch - this is used for calculating cumulative setup and hold times.
LTCHOUT	Output from a transparent latch - this is used for calculating cumulative setup and hold times.
CLKIN	Input to an edge-triggered latch - this is used for calculating cumulative setup and hold times.
CLKOUT	Output from an edge-triggered latch - this is used for calculating cumulative setup and hold times.
CLOCK	Clock input - this is used for calculating cumulative setup and hold times.
OE	Open emitter output - i.e. can pull up but not down.
NC	A no-connect pin

Appendix B - Primitive Device Types

Every device on a DesignWorks schematic has a characteristic known as its *primitive type*. The primitive type is set when the part entry in the library is created and cannot be changed for individual devices on the schematic.

Primitive types fall into three general groups:

• Schematic symbols: The two primitive types SUBCIRCUIT and SYMBOL fall into this category and are the normal primitive types used for creating schematic symbols. SUBCIRCUIT is the default type for symbols created using the DevEditor. There are no restrictions on the ordering or type of pins on these symbols.

• Pseudo-device types: These are the symbols used for page connectors, power and ground symbols, etc.

IMPORTANT: DesignWorks has very specific requirements for the order and type of pins on pseudo-devices. Refer to the following table for information. These rules ARE NOT CHECKED by the DevEditor.

• Simulation types: The majority of the primitive types defined in the following table are simulation primitives and are intended for use with the DesignWorks Digital Simulator option.

IMPORTANT: The simulation primitive types should not be used for user-created symbols without a clear understanding of their function. See the Digital Simulator manual for more information.

Schematic Symbol Primitive Types

Primitive Type	Pin Requirements	Comments
SUBCIRCUIT	No restrictions	Symbol having an optional internal circuit. This is the default for symbols created using DevEditor.
SYMBOL	ENo restrictions	Symbol with no internal circuit.

Pseudo-Device Primitive Types

IMPORTANT: The pin requirements listed in the following table must be followed when creating pseudo-device symbols. These rules ARE NOT CHECKED by the DevEditor.

Primitive Type	Pin Requirements	Comments
BREAKOUT	Pin 1 is Bus Pin, followed by N Normal Pins, set to Input	Splits signals out of or into a bus.
SIGNAL CONNECTOR	Exactly 1 normal pin, normally set to Input	Used for power and ground connections.
PORT CONNECTOR	Signals - exactly 1 pin, Busses- exactly 1 bus pin with any number of internal pins	Makes a connection between the signal it is connected to and a like named pin on the parent device.
PAGE CONNECTOR	Signals - exactly 1 pin, Busses- exactly 1 bus pin and no internal pins	Exports the signal or bus globally across all pages at one level of a schematic.

Simulation Primitive Types

WARNING: Do not use these without referring first to the Digital Simulation manual for more information!

Primitive Type	Pin Requirements	Comments
NOT	1 Input, 1 Output, ...	Output complement of Input. Same for each gate.
AND	N Inputs, 1 Output	Output <- AND of all Inputs
OR	N Inputs, 1 Output	Output <- OR of all Inputs
XOR	N Inputs, 1 Output	Output <- XOR of all Input
XNOR	N Inputs, 1 Output	Output <- XNOR of all Input
D-FF	/Set, Data, Clock, /Reset, Q, /Q	
JK-FF	/Set, J, /Clock, K, /Reset, Q, /Q	
D-FF (non inv)	Set, Data, Clock, Reset, Q, /Q	Set and Reset are active high.

JK-FF (non inv)	Set, J, /Clock, K, Reset, Q, /Q	Set and Reset are active high.
BUFFER	1 Input, 1 Output, ...	Output is Input. Same for each gate.
MULTI-PLEXER	N Data, M Select, /E, Output	Routes 1 of N inputs to the output. M is the smallest value such that $2M \geq N$
DECODER	N Select, M Output, /E,	Selects 1 of M outputs. N is the smallest value such that $2N \geq M$
REGISTER	N Inputs, N Outputs, Clock, Clear	Inputs are latched when clock is high, clear is async.
SHIFT REG	N Inputs, N Outputs, Clock, Shift/~Load, Shift-In	Shifts or loads data on clock.
ADDER	N A-Inputs, N B-Inputs, N Outputs, Carry In, Carry Out	Adds A and B w/Carry
COUNTER	N Inputs, N Outputs, Clock, Count/~Load	Counts up or loads on clock.
X-GATE	2 Bidirectional, Control, ...	The 2 bidirectional pins are connected if control is active. Repeated for each gate.
RESISTOR	2 Bidirectional, ...	Converts high drive to low drive for each resistor.
PULL_UP	1 Bidirectional, ...	Outputs a high with a low drive level for each pin.
ONE SHOT	Clock, /Clear, Q, /Q	Retriggerable. Delay is 1 by default. Delay setting is used as pulse width.
CLOCK	1 Output	The delay setting is used as the high and low period of the clock.
PROBE	1 Input	Displays Boolean input value on the schematic
SPST-SWITCH	2 Bidirectional Pins	Makes a connection between the two pins when switch is closed, otherwise both pins are undriven.
SPDT-SWITCH	3 Bidirectional Pins	Has pins A, B, and C; A and C are connected or B and C are connected.
HEX KEYPAD	Q0..3 Outputs, Strobe Output	Outputs a 4 bit hex value, strobe goes high when key pressed.
PUSH BUTTON	3 Bidirectional Pins	A, B, and C Pins; A and C are connected or B and C are connected. Switch is momentary.
SETUP & HOLD	Clock, 1 Data (Input), Error (Output)	Error = X if Data changed after Setup or before Hold time, Error = Z otherwise.

GLITCH DETECTOR	Sense (Input), Error (Output)	Error = X if sense changes faster then delay setting, Error = Z otherwise.
UNKNOWN DETECTOR	Sense, Output	If Sense = 0 or 1 then Output = 0, Output = 1 otherwise.
SIMULATOR STOP	1 Input	Performs requested task (stop simulator, generate report, ...) on rising edge.
LOGICBOX		Represents LogicBox hardware interface. See LogicBox Manual.

Appendix C - Predefined Attribute Fields

The following tables list fields that are predefined for each new design that is created. These fields cannot be removed because many of them are used by specific DesignWorks features.

For convenience, the fields marked as "primary" by default are listed first since they are most commonly entered directly by the user.

The following information is provided in the table:

Field Name	The name of the field as it appears in the Define Attributes box.
Used In	The types of objects this field is used in.
Instance/ Definition	Specifies whether the field data is kept with the definition or instance or a sub-circuit.
Description	How the field is used.

NOTE: Many of the predefined fields are used for specific DesignWorks features. More details on the usage of these fields can be found in the manual sections covering those features.

Primary Fields

The following fields are the ones most commonly edited directly by the user.

Field Name	Used In	Inst/Def	Description
CctName	Design	Def	Design file name. Sets the window title and name of next saved file.
Description	Devices	Def	A short description of the device type. This is included in all standard DesignWorks libraries. Not used internally.

Ground	Devices	Def	List of pins to be hooked to ground net in netlist, separated by commas. Format example: `7,10,13`
InstName	Devices, signals	Inst	Name applied to physical instance. This is normally used for the package name in hierarchical designs. Normally not used in Flat or Pure mode designs.
Name	Devices, signals	Def	Definition name, i.e. reference designator. This is the field set using the text tool on the schematic.
Package	Devices	Def	Device package code. This is included in all standard DesignWorks libraries. Not used internally.
Part	Devices	Def	Part type code. Used by Packager to assign gates to packages.
Power	Devices	Def	List of pins to be hooked to power net in netlist, separated by commas. Format example: `1,2,14`
PrefixField	Design	Def	Contains the name of the field to use as a name prefix for device packaging and device auto-naming.
Spice	Devices, Design	Def	Holds simulation parameters for SPICE-based simulators. Not used internally.
Unit	Devices	Inst	Gate unit, e.g. a, b, c, etc. Set and checked by Packager or can be set manually.
Value	Devices	Def	Component value to appear in netlists. Not used internally.

Secondary Fields

In general, values in the secondary fields should not be edited directly by the user without a thorough understanding of their effect.

Field Name	Used In	Inst/Def	Description
AutoSym.Bottom	Devices	Def	The string used in the "bottom pins" box in the Auto Create Symbol command in the DevEditor.
AutoSym.Left	Devices	Def	The string used in the "left pins" box in the Auto Create Symbol command in the DevEditor.
AutoSym.Right	Devices	Def	The string used in the "right pins" box in the Auto Create Symbol command in the DevEditor.
AutoSym.Top	Devices	Def	The string used in the "top pins" box in the Auto Create Symbol command in the DevEditor.
BusInfo	Pins	Def	Used only on bus breakouts to notate the names of the attached signals. Normally only updated using the Bus Pin Info command.
Category	Devices	Def	Contains a terse device category code. Can be used as an alternate device prefix by entering the name of this field in PrefixField. Not used internally.
CctPath	Design	Def	The directory path name for the design file. This is used only for creating external circuit references for sub-circuit parts. Changing it will not affect the current design file.
Depth	Devices	Def	Indicates nesting depth of device sub-circuit. Only used temporarily by the Report tool while generating hierarchical reports. This value is not maintained during editing operations.
DevPrefix	Design	Def	Default prefix for auto-assigned device names. Normally set using the Design Preferences command.

ExtCctName	Devices	Def	Name of external design file containing device internal circuit. Normally set by the DevEditor.
ExtCctPath	Devices	Def	Directory path for design file containing device internal circuit. Normally set by the DevEditor.
Function	Devices	Def	Contains a terse device function code. Can be used as an alternate device prefix by entering the name of this field in PrefixField. Not used internally.
HierNameSep	Design	Def	Contains the separator string used in generating hierarchical device and signal names for Report output. Normally set only by commands in a report form file.
LibDate	Devices	Def	"Last Modified Date" of the library the device was read or updated from, stored in encoded form. This is used by the Update from Lib command.
LibName	Devices	Def	Name of the library the device was read or updated from. This is used by the Update from Lib command.
LibPath	Devices	Def	Directory path of the library the device was read or updated from. This is used by the Update from Lib command.
LibType	Devices	Def	The original name of the part when it was read from a library. This is used by the Update from Lib command.
Name.Prefix	Devices	Def	Prefix used to create default Name value. Used by Packager and Auto-name.
Name.Pt	Devices	Def	Default position of Name attribute relative to device.
Name.Spice	Devices	Def	Alternate prefix for SPICE-based simulators.
OKErrors	Devices, Signals, Pins	Def	Used by ErrorFind tool to mark errors set as OK by user. Should not be set manually.

Package.List	Devices	Def	List of package types that apply to this part type.
PageRef	Devices	Def	Used in Page Connector pseudo-devices only. Contains the auto-generated page references.
PageRefFormat	Design	Def	Format string for page references. Normally modified only using the Design Preferences command.
PageRefWidth	Design	Def	Max. number of items per line in page references. Normally modified only using the Design Preferences command.
Part.List	Devices	Def	List of part types possible for this symbol.
Part.Pt	Devices	Def	Default position of Part attribute relative to device.
Permutable	Devices	Def	This field contains a list of the swappable pins and gate sections in this part type. Parentheses "()" enclose items that can be swapped, while brackets "[]" enclose items that cannot be swapped. This is intended for use by external PCB layout packages. Not used internally.
PinSequence	Devices	Def	This field is used to determine pin order in netlist output formats that require specific device pin ordering, e.g. SPICE. This is not defined in the standard DesignWorks libraries. Not used internally.
PkgLevel	Devices	Def	An integer 0 to 3 specifying packaging action: 0 = normal 1 = lock and check 2 = lock and don't check 3 = ignore Normally set using the Device Info or Get Info commands.
PkgPrefix	Design	Def	Default prefix used by Packager to create a package name. Normally set using the Default Prefix command.

Restrict	Devices	Def	An integer value from 0 to 7 used to set access restrictions for the device's internal circuit. The value is the sum of the following: 1 = Don't report 2 = Don't package 4 = Don't Push Into Normally set using the Get Info or Device Info command.
SigPrefix	Design	Def	Prefix used to generate default signal names. Normally set using the Design Preferences command.
Unit.All	Devices	Def	List of all units available in this part type. Used by Packager.
Unit.List	Devices	Def	List of all units in this part type having this symbol. Used by Packager.
UnusedPins	Devices	Def	List of unused pins in this part type. Not used internally.
VisPin.List	Pins	Def	List of pin numbers for this pin corresponding to the gate units in "Unit.List". Used by Packager.

Appendix D - Setup File Format

The first word on each statement is a keyword which specifies a setup option. Statements are terminated by a semi-colon and can contain embedded comments in braces, as in `{ Comment }`. DesignWorks recognizes the following keywords:

Library and Circuit Folder

```
FOLDER folderName;
```

This specifies the folder that will contain the libraries to be specified in following `LIBRARY` statements, or circuit files for the following `CIRCUIT` statements. This statement can be omitted if the files are in the same folder as DesignWorks, or if you prefer to specify a complete file "pathname" in each library statement. If the name begins with a colon the named folder is assumed to be in the same folder as DesignWorks, otherwise the name is taken to be the name of a disk. The FOLDER option does not actually change what the system thinks is the "current" folder, it just specifies a prefix to be added to following library names. You can therefore specify multiple FOLDER statements, as in the example later in this section. Each one is valid until the next FOLDER statement. The following are valid statements:

```
FOLDER :Libs:;
```

specifies a folder called Libs contained in the same folder that contains DesignWorks.

```
FOLDER LibDisk:4000 Series:;
```

specifies a folder called "4000 Series" on a disk called LibDisk.

```
FOLDER :Libs:7400 Counters:;
```

specifies a folder called "7400 Counters" which is contained inside a folder called Libs which is in the same folder (or on the same disk) as DesignWorks.

Tools Folder

```
TOOLFOLDER folderName;
```

The TOOLFOLDER keyword is used to specify the folder to search for MEDA modules to load when the program is started. All modules in the specified folder will be loaded. The "folderName" has the same format as for the FOLDER statement described above.

Default Libraries

Single Libraries

```
LIBRARY libName;
```

This specifies the name of a library to open. The library will be opened each time DesignWorks is started up and its name will appear in the Parts palette. As discussed above, libName can be a simple file name or a complete pathname specifying the disk and folders containing the library. Note that the name specified in the last FOLDER statement is always attached to the front of the name specified in the LIBRARY statement, even if a complete pathname is given. Following are examples of acceptable LIBRARY statements (these all assume no FOLDER statement has been used):

```
LIBRARY LibDisk:Counters;
```

specifies a library called Counters on a disk called LibDisk.

```
LIBRARY Discretes;
```

specifies a library called Discretes in the folder DesignWorks is in (or in the folder specified by the last FOLDER statement.

All Libraries in a Folder

```
LIBRARYFOLDER folderName;
```

The LIBRARYFOLDER keyword names a folder to be searched for libraries. All libraries in this folder will be opened. Folders inside this folder are not checked. The format of the folder name is same as for the FOLDER keyword above.

Default Design

```
DESIGN circuitName;
```

This statement allows you to specify a circuit file to open when the program starts. This can be used to open a file that is being repeatedly edited, or to open a default "template" file with a standard title block or border. NOTE: If the circuit is in any folder other than the one containing the DesignWorks program, you must use the FOLDER statement to specify the folder name. Unlike the LIBRARY statement, DESIGN does not allow a complete file "pathname" to be specified.

Color Settings

```
COLOURS colourList;
```

or

```
COLORS colorList;
```

(depending on whose dictionary you use)

This specifies colors to be used in displaying objects on the Macintosh II screen. The statement has no effect on other machines. Each number in the list corresponds to a different type of circuit object. The COLOUR keyword is followed by a number of integers which will determine the colour displayed on the screen for various types of objects. If less numbers then object types appear in the COLOUR statement, the remaining ones are left unchanged from the defaults, which are specified in a resource entry called CTAB in the DesignWorks file. Extra numbers on the line are ignored.

Valid colour numbers are determined by the Macintosh system, and must be from the following list:

BLACK
WHITE
RED
GREEN
BLUE
CYAN
MAGENTA
YELLOW

Due to a quirk in the Macintosh system, device pictures can only be displayed in the colour they were originally created in, which is usually black. The overhead required to overcome this problem would considerably slow down screen updates, so we have not implemented it.

These colors are read in the following order:

1	Default colour for items not specified
2	Name on a selected signal
3	Name on an unselected signal
4	Pin numbers on selected signal
5	Pin numbers on unselected signal
6	Reserved
7	New internally-created device symbols
8	Pin line in selected signal
9	Pin line in unselected signal
10	Non-pin line in selected signal
11	Non-pin line in unselected signal
12	Delay in selected device
13	Delay in unselected device
14	Name in selected device
15	Name in unselected device
16	Drawing boundary
17	Drawing grid
18	Print page outlines
19	Random text
20	Frame on random text
21	Rulings on random text
22	Frame on random pictures
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved
32	Reserved
33	Reserved
34	Reserved
35	Reserved
36	Reserved
37	Reserved
38	Pin line in selected bus
39	Pin line in unselected bus
40	Non-pin line in selected bus
41	Non-pin line in unselected bus

IMPORTANT NOTE: The SCALES, NORMALSIZE, PINSPACE, SIZES, STYLES, FONTS, BREAKOUT and NOERRORCHECK items following are provided for experimental use by experienced users of DesignWorks. The program is optimized for the default settings and is not guaranteed to adjust correctly to all possible changes. Test out the system thoroughly with any new settings before performing any important tasks.

Zoom Factors

Reduce/Enlarge Settings

```
SCALES n1..n11;
```

The SCALES keyword is used to specify the magnification levels used by the Reduce and Enlarge commands. The keyword is followed by 1 to 11 decimal integers separated by blanks and sorted in ascending order. The 1:1 scale level (at which externally created pictures appear in their original size) is 14. Enlargements are specified by smaller numbers (e.g. 7 gives 200%) and reductions by larger numbers. Since all scaling is done using integer arithmetic, the best results are obtained at scale factors which are multiples or factors of 14. The default values are:

```
SCALES 4 7 10 14 18 24 28 42 63 98 140;
```

Normal Size Setting

```
NORMALSCALE n;
```

The NORMALSCALE keyword is used to specify which of the scale steps specified in the SCALES line will be used as the "Normal Size" setting. The keyword must be followed by a single decimal integer in the range 1 to 11 which is the ordinal number of the scale step to be used as normal. The default is 4.

Pin Spacing

```
PINSPACE n;
```

The PINSPACE keyword is used to specify the spacing between adjacent pins when breakout symbols are created. This can also serve as the default for symbols created by other tools. The keyword must be followed by a

single decimal integer which is a multiple of the standard grid space of 5 dots. The default value is 2.

Breakout Parameters

```
BREAKOUT dth dtv;
```

The BREAKOUT keyword lets you control the creation of bus breakout symbols generated by the program. This does not affect any breakouts in existing files as these symbols are already created and stored with the file.

The BREAKOUT keyword is followed by six numbers for the following parameters:

dth	the horizontal offset (in screen dots at 100% scaling) for placement of text names on a breakout.
dtv	the vertical offset (in screen dots at 100% scaling) for placement of text names on a breakout.

Internal Error Checking

```
NOERRORCHECK;
```

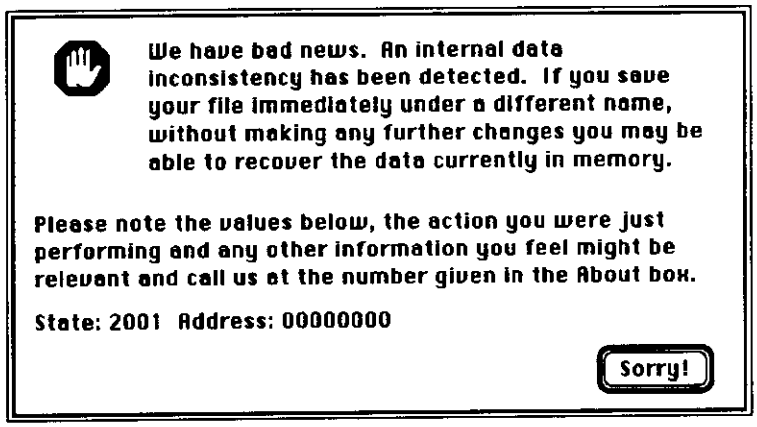
This option disables the internal error checking described elsewhere in the Appendix.

NOTE: This should only be done in consultation with our Technical Support department as it will prevent warnings from being issued for internal or unusual system errors. In many cases these errors can be easily recovered if corrected immediately but may cause data corruption if left undetected.

Appendix E - Internal Error Detection

DesignWorks makes use of a number of complex internal data structures in order to maintain an up-to-date image of your circuit at all times. To assist in detecting problems due to hardware failures, program errors or operating system errors, a code module has been added which checks these structures for consistency. This is done in the "background" while the program is idling and should normally be invisible to the user.

Should a problem be detected, a warning box similar to the following will be displayed:


We have bad news. An internal data inconsistency has been detected. If you save your file immediately under a different name, without making any further changes you may be able to recover the data currently in memory.
Please note the values below, the action you were just performing and any other information you feel might be relevant and call us at the number given in the About box.
State: 2001 Address: 00000000
Sorry!

The "State" value is a code that specifies the type of problem detected, and the "Address" value is the memory address of the object that was in error. It is beyond the scope of this manual to discuss the meaning of all possible error codes and an estimate of their severity. In general, a State < 100 indicates a structural problem that is likely to cause a serious program malfunction if you proceed with editing. The warning box will only appear once if this type of error is detected, even if other errors occur later. The error detection mechanism is reset when all design files are closed. States > 100 indicate unexpected situations detected in connection with some specific function and may or may not be serious. If the error disappears after a later check (due to an offending object being deleted, for example) you will be notified. This may occur if you delete the corrupted part of the circuit or if some other internal check succeeds in correcting the problem.

If you see this box in the course of normal program operation, then save your design file immediately under a different name (so as not to wipe out your last good backup). Quitting the program and rereading the saved file may result in the problem being corrected. If you can isolate the problem

down to one specific object, then try deleting that object and recreating it. In any case, please contact our Technical Support department and provide as much information as you can about the situation which created the problem. We will help you in any way we can to recover any lost data.

Error checking can be disabled using the NOERRORCHECK setup file option, described elsewhere in the Appendix. This should not be used under normal circumstances.

Credits

DesignWorks, LogicWorks, and MEDA are trademarks of Capilano Computing Systems Ltd.

DesignWorks was designed by Chris Dewhurst and coded by Chris Dewhurst, Don Gamble, Geoff Harris, Addam Smith, David Taylor and Neil MacKenzie of Capilano Computing Systems Ltd.

This manual was written by Chris Dewhurst using Microsoft Word (a trademark of Microsoft Corporation of Bellevue, Washington).

DesignWorks is written in the C and Modula-2 languages under the MPW system.

Macintosh is a trademark of Apple Computer, Inc. of Cupertino, California.

Other trademarks used in this manual are property of their respective holders.

Technical Support

Any problems or questions can be referred to:

Technical Support,

Capilano Computing Systems Ltd.,

406 - 960 Quayside Drive,

New Westminster, B. C.

Canada, V3M 6G2

Voice (604) 522-6200

Fax (604) 522-3972

AppleLink: D0816

Bulletin Board System

The DesignBoard BBS is available especially for our customers to access technical support, download updates and technical notes, exchange files and communicate with Capilano personnel.

Access number: (604) 522-4361

Access parameters: 2400 Baud, 8 bits, no parity

On your first access you can use basic functions and request validation, or you can phone us in advance and we will create an ID for you.

Index

Page numbers in **BOLD** face indicate major references to a topic.

$ALIGNCOLSON/OFF **273**
$ASCENDING 285
$ASSIGNNAMES **282**
  example 249, 257, 260, 262, 263
$ATTRNUM **275**
$AUTONUMBER 33, 81, **271**
  example 249, 260
$BLANKREPLACE **274**
  example 263
$BUSNAME 293
  example 262
$CHANGECOUNT 297
$CHECK 245, **290**
  example 254
$CHILDSIGNAME 279
$CIRCUIT **299**
$CLOSEREPORT **277**
$COL
  after DEVICES 296
  after SIGNALS 294
  example 249
$COMBDEVSON/OFF **287**, 292, 297
  example 250, 260
$COMBPINSON/OFF **272**
$COMBSIGSON/OFF **287**, 292
$CONTEND **274**
  example 263
$CONTREPEATON/OFF **275**
$CONTSTART **274**
$COUNT
  after DEVICES 295
  after SIGNALS 293
$CREATEREPORT 243, **276**
  example 257, 260
$DATE
  example 257
  in body 297
  in header 282
$DATECREATED variable 74
$DATEMODIFIED variable 74
$DATENOW variable 74
$DEPTH
  in SORT 286
$DESCENDING 285
$DESIGNNAME
  example 257
  in body 297
  in CREATEREPORT 276
  in header 282
$DEVCOUNT 297
$DEVHIERNAME 303
  in pin format 279
  in SORT 285
$DEVICES 271, 278, **295**, 299
  example 250, 257, 261
$DEVINSTNAME 303
  in pin format 279
  in SORT 286
$DEVNAME 303
  after DEVICES 295
  in FIND 289
  in pin format 279
  in SORT 285
$DEVPINFORMAT **278**, 296
  example 261
$DEVPINSEQUENCE 131, **271**, 296
$DEVSEQ
  after DEVICES 295
  in pin format 279
$DEVTOKEN
  after DEVICES 295
  in pin format 279
$ENDHEADER 282
$FILENAME variable 75
$FIND **288**
  example 254
  patterns 290
$FLAT 283, 291, **299**
$FLATDOWN 283, **299**
$GRID

after DEVICES 296
in pin format 279
$HEADER **282**
$HIERARCHY 272, 286
$ITEMSEPARATOR **273**
$LINESUSED 269, 297
$LINEWIDTH **269**, 274
$MAXITEMSPERLINE 269, **270**, 274
example 264
$NEWLINE 292
after DEVICES 296
after SIGNALS 294
example 260
in pin format 279
$NEWPAGE 297
$NOT 289
example 254
$NOTES 241, **281**
$NUMINPS 296
$NUMOUTS 296
$NUMPAGES variable 74
$NUMPINS
after DEVICES 296
in FIND 290
in SORT 286
$ONEPINSON/OFF **274**
$ORIENT
after DEVICES 295
in SORT 286
$PAGE
after DEVICES 296
after SIGNALS 294
in header 282
in pin format 279
in SORT 286
$PAGELENGTH **269**, 297
$PAGENUM variable 74
$PAGETITLE variable 74
$PHYSICAL 291
$PINCOUNT 297
$PINDIR
in pin format 280
$PINNAME
in pin format 280
$PINNUM
in pin format 280
$PINS **294**
after DEVICES 296
example 257
$PINSEQ
example 262
in pin format 280
$PINTYPE
in pin format 280
$PINTYPEFORMAT **278**, 280
$POSX
after DEVICES 295
in pin format 280
in SORT 286
$POSY
after DEVICES 295
in pin format 280
in SORT 286
$PRINTNUMPAGES variable 75
$PRINTPAGENUMBER variable 74
$PURE 286, 291, **299**
$PUREDOWN **299**
$REPEATITEMON/OFF **275**
$SELECTED
in FIND 289
$SIGCOUNT 297
$SIGHIERNAME 303
in pin format 280
in SORT 286
$SIGINSTNAME 303
in pin format 280
$SIGNALS 278, **293**
example 248, 264
$SIGNAME 303
after SIGNALS 293
in FIND 290
in pin format 280
in SORT 286
$SIGPINFORMAT **278**, 294
example 248
$SIGSEQ
after SIGNALS 293
example 262
in pin format 281

$SIGSOURCE 77, 79, 243, **270**
example 249, 260
in FIND 290
in SORT 287
$SIGTOKEN
after SIGNALS 293
in pin format 281
$SINGLE
after DEVICES 295
after SIGNALS 294
$SORT **285**
example 249, 261
$SPACE **273**
$SYSPIN
in pin format 281
$TIME
example 257
in body 297
in header 282
$TIMECREATED variable 74
$TIMEMODIFIED variable 74
$TIMENOW variable 74
$TOPCIRCUIT **299**
$TYPENAME
after DEVICES 295
in FIND 289
in SORT 286
$UNCONNPINSON/OFF **275**
$UNNAMEDDEVS **273**
$UNNAMEDSIGS **274**
&Attribute variables
in Report 268
in text 75
.List fields **97**, 189, 193, 197
.Pt fields **98**, 236, 237
Add Bus Sigs button **196**
Add Pins command 206, 207, 215, 227
Align commands 215
Allow Carriage Returns option 101, 179
Attach Internal command 122, 125, **175**
Attribute Probe command 23, 94, 101, 106, **149**
attributes **93**
Allow Carriage Returns 101
default position fields **98**
default value 99
Delete function 180
design 75, 105, 164, **180**, 257, 263, 276, 277, 282, 298
devices 42
Duplicate function 180
editing on schematic 148
find by value 319
Group with Name option 100
hiding 200
in DevEditor 227
instance 100
Justification command 199
list menu 189, 193, 197
maximum length 102, 180
package type 80
power and ground 76, 242, 270
predefined fields 329
primary **94**, 101, 329
primary fields 179
Rotate Text With Object option 101
rotation 106, 201
secondary 331
Show Field Name 100
Show Field Name command 201
text style **157**
value list fields **97**
variables 75
Visible by Default option 101
Attributes command 42, 95, 105
devices **187**
pins **195**
signals **190**
attributes in Report 268
devices 265, 296
in FIND 289
in header 282
in report body 298
in SORT 286, 287
power and grounds 242, 270
signals 266, **294**

Auto Create Symbol command 120, 124, 207, 215, **229**
Auto Packaging Enabled command 56, **184**
automatic page references **67**, 156
AutoSym.xxx attribute fields 331
background grid **159**
  minor divisions 160
background objects **73**, 83, 147, 150
  selecting 49
backspace key 146
behavioral model 230
bidirectional pin 210
border 36, **83**, 159
  text style 160
border objects 36, 52, **73**, 76, 171, 172
breakout device 32, 69, **70**, 326
  custom 232
  setup file parameters 340
Bring to Front command **214**
Browser tool 38, 41, 49, 57, 254, 282, **309**
bus page connector 32, 68
Bus Pin Info command 72, 87, 90, **195**, 331
BusInfo attribute field 331
busses 68, **69**
  breakouts 232
  bus pins 28, 69, **72**, 87, 90, **117**, 120, 157, 169, 192, 195, 206, 210, 228, 324
  creating 87
  Page Connector 234
  pin annotation 197
  Port Connector 233
Category attribute field 331
CctName attribute field 329
CctPath attribute field 331
Center in Page command 52, 73, **161**, 171, 172
circuit
  definition 28
Circuit Info command 198
Clear command 42, **146**, 189, 193
  attributes **200**
clipboard 35, **37**, **144**
  in dialog boxes 18
Close Design command **135**
Close Lib command **140**
Close Part 217
Collect pins option 212
COLORS setup file keyword 337
colour 15, 337
columns in Report
  alignment 273
  line width limit 269
  spacing 273
command key 33, 35, 43, 97, 152, 199
comments
  in Report 268
Compact library option 141
connectors 80
  page connector 66
  power and ground 78
  signal connector 78
continuation lines 275
continuation lines in Report 274, 275
Converter tool 15
Copy command 18, **145**, 188, 192
  DevEditor 217
  library parts 141
Copy Pin Names command 216
cursor position display 21
Custom Sheet Info command 26, 35, 84, 128, 136, 157, **158**
  border text 91
Cut command **145**, 188, 192
  DevEditor 217
date and time variables 74
default border 159
  text style 160
default names
  assigning 282
  auto-naming 55, **64**
  example 249
Default Prefix command **184**, 333
Define Attribute Fields command 58, 94, 99, 103, 107, 150, **178**, 271, 313
delete key 42, 44, 146

Delete Page option 152
Delete pins option 212
Demote
  library part 141
  page 153
  pin 211
Depth attribute field 286, 331
Description attribute field 329
design
  structure 27
Design Attributes command 130
Design Preferences command **155**, 331, 333
  auto-naming 55, 64
  cursor position display 21
  free memory display 21
  hierarchy mode 119
  page reference options 67, **91**, 192
  Show Page Breaks 137
  Show Printed Page Breaks 137
  text style 107
  text styles 90
DESIGN setup file keyword **337**
Detach Internal command **177**
DevEditor tool 29, 33, 53, 58, 69, 77, 79, 80, 107, 131, 140, **203**
  Auto Create Symbol command 229
  creating a new part 203
  editing an existing part 203
  hierarchical blocks 120
  pin tools 206
  port names 116
Device Info command 187
devices
  auto-naming 33, 55, **64**, 155, 235
  connectors 80
  definition 28
  discrete components 80
  libraries 29, 30
  moving 42
  name 96
  naming 45
  package type 80
  packaging **55**, 96, **112**, 184, 234
  pin numbers 148
  pin type **323**
  pop-up menu 187
  primitive type 231, **325**
  pseudo-devices 325
  report options 246
  rotation 162
  symbol creation 203
DevPrefix attribute field 331
discrete commponents
  name prefix 236, 283
  pin numbers 166, 169, 249, 260, 271
  pin type 323
  SPICE 92, 131
  Value attribute field 237, 255
discrete components 80
  name prefix 59
  package types 82
  pin numbering 81
Draw Bus command 22, **149**
Draw Sig command 23, **149**
Drawing menu 151
Duplicate command **147**, 189, 192
  attributes **200**
Edit command **199**
Edit Part command 140, 225
Enlarge command **153**, 154, 198, 339
error detection (internal) 340, 341
ErrorFind tool 31, 38, **317**
escape character 261
Excel
  exporting to 252
ExtCctName attribute field 332
ExtCctPath attribute field 332

Extract Pin List button 230
fanout 321, 323
file name variable 75
Fill Down command 314
fill patterns 205
Fill Right command 314
Find tool 38, 40, 96, **317**
flat hierarchy mode **112**, 127
  names 95, 186
  netlists 242, 298
  packaging 55, 61, 64, 96, 156
Flip Horizontal command 188
Flip Vertical command 188
font 157
  attributes 90
  border 91
  DevEditor 216
  pin numbers 90
  text blocks 51
Font menu 216
form file
  format examples 247
  language reference 265
  naming convention 242
Format Notes option **241**, 281
free memory display 21
Free Unit Report command **185**
Function attribute field 65, 92, 332
gate packaging 55
Generate button 230
Get Info command **163**, 187, 246, 270, 300
  attributes 42
  breakout pins 89
  busses 70, 71, **168**
  circuits **163**
  design **163**
  devices 42, 105, 138, **165**
  in Finder 15
  packaging options 58, **59**, 60, 62
  page **163**
  page connectors 166
  pictures 37, 73, 83, 85, 144, 171
  pins 28, 48, 81, 131, 169
  pseudo-devices 167
  selection 38
  signals 47, **167**
  text 51, 73, 82, 85, 91, **170**
Go To Selection command **151**
graphics files 136
graphics objects
  pasting 37, 144
grid 339
  background 4, 35, 136, 159
  cursor position 21
  DevEditor 213
  reference 76, 84, 91, 156, 296
Grids command **213**
ground and power connections **76**, 78, 232, 242, 270
  in hierarchy 116, 118, 304
Ground attribute field 237, 242, 249, 270, 330
Group command **214**
Group with Name option 100, 179
Hide command **200**
Hide Tool Palette command **150**

hierarchy **109**
definition vs. instance 111
example 109
flat mode **112**
in Report 298
mode 112
mode check in netlist 291
mode selection 156
physical mode 97, **112**
pure mode 97, **112**
HierNameSep attribute field 126, 332
ImageWriter 137
Import Sheet Info command 35, **39**, 52, 66, 73, 76, 158, 171, 172
In Primary List option 101
input pin 210
installation 11
serial number 12
setup file 11, 15, 335
instance data 100
Instance fields 94
InstName attribute field 61, 64, 95, 96, 283, 330
internal circuit
port interface 211
Join bus pin option 195
Join Sequential bus pin option 195
Justification command **199**
Keep with Instance option 100, 179
LaserWriter 137
Lib Maint command 140
LibDate attribute field **182**, 332
LibName attribute field **182**, 332
LibPath attribute field **182**, 332
libraries 29
automatic opening 336
compaction 141
creating 30, **139**
maintenance 140
Libraries sub-menu 139
LIBRARY setup file keyword **336**
LIBRARYFOLDER setup file keyword **336**
LibType attribute field **182**, 332
Lock opening internal circuit 138, 223
logical name 96
Magnify command 23, **154**
Magnifying command 40
Make Invisible command 314
Make Unique Type command **181**
Make Visible command 314
Mark Current OK option 322
MEDA **8**, 18, 29, 230
search path 336
memory available display 21
memory usage 15
MultiFinder 15
Name attribute field **95**, 96, 330
auto-naming 64
busses 32
default position 236
device 148
packaging 156
signal 148
signals 32
Name command 46
devices 188
signals 190
Name.Prefix attribute field **58**, 62, 65, 92, 236, 332
Name.Pt attribute field **98**, 236, 332
Name.Spice attribute field 65, 92, **130**, 236, 332
names
auto-naming 33
busses 32
device 96
editing 47
hierarchical names 126
invisible 79, 95
moving 47
Name vs. InstName **95**, 96
pin names 207
pins 209
prefix field 92
removing 47
signals 32
netlists 65, **239**
hierarchical 113, 298

New Breakout command 69, 87, **88**, **172**, 196
New Design command 38, **134**
New Lib command **139**
New Page option 152
New Part command 216
NOERRORCHECK setup file option 340, 342
Normal Size command **153**, 198, 339
NORMALSCALE setup file keyword 339
OKErrors attribute field 332
open collector pin 210
Open Design command **135**
open emitter pin 210
Open Lib command 29, **139**
Open Page option 152
Open Part command 217, 225
Option key 34, 41, 43, 50, 51, 146
Orientation command 21, **162**
  Paste command 146
output pin 210
Package attribute field 237, 330
Package.List attribute field 237, 333
packaging **55**, 86, 96, 97, 156, 184, 234, 235
  default prefix 184
page
  creating a new page 152
  definition 36
  deleting 152
  opening 152
  page number variables 74
  page title 153
  PAGETITLE variable 74, 153
  printer page setup 137
  references 67
  reordering 153
  title block 82
Page command 19, 66, 73, 74, 82, **151**
Page Connector device 28, 31, **66**, 96, 156, 166, 234
  page reference format 160
  unmatched 321
page header 282
page references 156, 333
  format 91, 156
Page Setup command **137**, 160
PageRef attribute field 333
PageRefFormat attribute field 333
PageRefWidth attribute field 333
Pages command 198
paper size 28, 36, 137
Part attribute field 62, 97, 165, 236, 247, 330
Part Attributes command 107, 215, **227**
Part.List attribute field 236, 333
Part.Pt attribute field **98**, 237, 333
Paste command **145**
  auto-connection 145
  DevEditor 217
  rotation 20, 21, 146, 162
  text 29, 35
  title blocks 83
patterns 290
Permutable attribute field 333
physical hierarchy mode **112**, 156
  Keep With Instance option 100
  names **97**, 114
  netlists 242, **299**
  packaging 55, 61, 64
  SPICE 130
physical hierarchy modes
  packaging 60
physical name 96
PICT clipboard data 37, 145
PICT file
  creating 136
  grid settings 159, 160
pictures 29
  background 73, 83
  Get Info 37, 144
  pasting 37, 144
Pin Attributes command 215, **227**
Pin Info command 169, 194
Pin List command 160, 167, 168, **191**
pin names 223

pin numbers **33**
auto-incrementing 34
default 210
default pin numbers 33
editing 148
rotation 158
text style 158
pin order in Report 130, 271
pin spacing
breakouts 174, 339
pin type **323**
pins
adding to symbol 218, 228
attributes 170
bus 210
bus internal 87, 192, 195, 210, 220, 223, 228
bus pins 28, 69, **72**, 87, 90, **117**, 120, 157, 169, 192, 195, 206, 228, 324
definition 28
DevEditor 206
error checking 321
input 323
inverted 229
normal 210
output 323
pin name 209
pin number **33**, 210
pin type 117, 210, **323**
placing on symbol 206
selecting 50
PinSequence attribute field 131, 272, 333
PINSPACE setup file keyword 339
PkgLevel attribute field 62, 333
PkgPrefix attribute field 62, 63, 333
plotting 28, 136
Point command 22, **147**
Pop Up command **174**, 198
pop-up menus 20
attributes 199
devices 187
orientation 21
Parts palette 25, 26
pins 194
signals 190
value lists 105
Port Connector device 28, 96, **116**, 211, 233, 321
error checking 321
port interface 211, 220
name matching 116
port pin type 117
unmatched ports 321
power and ground connections **76**, 78, 242
$SIGSOURCE 270
in hierarchy 116, 118, 304
Power attribute field 237, 242, 249, 270, 330
power net 232
predefined fields 329
prefix field 64
PrefixField attribute field **58**, 63, 65, 92, 130, 330
primary fields **94**, 101, 149, 179, 329
primitive type 166, 226, 231, **325**
Print Design command **137**
page numbering 74
Print Range command 137
printing **137**
ImageWriter 137
LaserWriter 137
paper size 28
Promote
library part 141
page 153
pin 211
pseudo-devices 28, 29, 64, 155, 170, 192, 325, 333

pure hierarchy mode **112**, 156
names **97**
netlists 113, 242, **298**
packaging 55, 60, **62**
restrictions 128
SPICE 130
Push Into command 118, **174**
devices 188
Reassign Device Names command **186**
Redo command **144**
Reduce command 40, **153**, 154, 198, 339
Reduce to Fit command 40, **153**, 154, 198
Repackage Design command 56, 57, **185**
Report tool 96
$ commands 267
attributes 42, 268
form file examples 247
format notes 241
hierarchical netlists 298
operation 244
pin numbering 81
power and ground 76, 270
signal names 32, 293
standard formats 241
Rescan Design command 56, 57, 60, **185**
Restrict attribute field 334
restricted opening 138
Rotate Left command 188, **201**
Rotate Right command 188
attributes **201**
Rotate Text With Object option 101
rotation
attribute text 101
devices 162, 188
Paste command 146
pin numbers 158
text 179
Save Design As command **135**
Save Design command **135**
Save Page Pict command **136**
Save Part As command 217, 224
Save Part command 217, 226
Save to Lib command **183**
SCALES setup file keyword 339
screen scaling **153**
secondary fields **94**, 331
Select All command **150**
selecting data in Report **288**
selecting objects **49**, 150
ErrorFind 322
Find 319
selection patterns in Report 290
Send to Back command **214**
serial number
entering 12
re-entering 13
Set Delay command 215
Set Design Attributes command 65, 75, 92, 126, **180**, 257, 264
setup file 29, **335**
DESIGN item **337**
format 335
installation 11, 15
LIBRARY item **336**
LIBRARYFOLDER item **336**
TOOLFOLDER item **336**
sheet 28, 134
definition 36
sheet info palette 158
shift key 33, 34, 42, 43, 49, 147, 152
DevEditor 208
Show Bus Pin Annotation option 197
Show Field Name command 101, **201**
Show Field Name option 100, 179
Show Parts Palette command 25
Show Printed Page Breaks option 137
Show Tool Palette command 21, **150**
signal connector device 33, 77, 78
creating 232
Signal Info command 190, 197

signals
auto-naming 64, 155
connecting by name 31, 33, 79, 95, 96
creating 30
definition 28
error checking 321
Get Info command 167
Name command 190
naming 31, 46
Pin List command **191**
pop-up menu 190
removing 44
selecting 49
sequential naming 34
SigPrefix attribute field 334
Size menu 216
sorting
Browser 315
in Report 284
SPICE **130**, 332, 333
netlist format 256
Spice attribute field **130**, 330
spreadsheets
exporting to 252
Style menu 216
subcircuit 166, 211, 226, 315, 325
DevEditor 220
internal/external 226
locking 166
Subcircuit & Part Type command 123, **215**, 221, 231
System 7.0 15
tabs in report 252
text **35**, **51**
background 52, 73, 171, 172
border 52, 73, **160**, 171, 172
editing 35, 148
pasting 37, 144
rotation 179
selecting 49
style
attributes 27, 90, **106**, **157**, 200
border 35, 91, **160**
DevEditor 216
pin numbers 90, 158
text blocks 51, 91, 171
variables 73
Text command 23, 45, **148**
title blocks 29, 73, **82**, 203, 337
tokens 279
devices 265
signals 266
tool palette **20**
TOOLFOLDER setup file keyword **336**
trademarks 343
tristate pin 210
type name 30, 79, **96**, 129, 165, 192, 229
in Report 254, 266, 283, 287, 295
Undo command **143**
Ungroup command **214**
Unit attribute field 56, 57, 58, **61**, 97, 189, 235, 330
Unit sub-menu 189
Unit.All attribute field 63, 235, 334
Unit.List attribute field 63, 235, 334
Unmark Selected option 322
UnusedPins attribute field 237, 334
Update from Lib command 27, 181, 332
Use Default Value button 99
Value attribute field 237, 247, 330
value list fields **97**, 105, 107, 189, 193, 197
variables
in Report 268
in text **73**
Visible by Default option 58, 101, 179
VisPin.List attribute field 63, 235, 334

Window menu 19
windows
    DevEditor 204
    Schematic 18
Zap command **23**, 37, **44**, 45, 47, 144, 148
zoom
    DevEditor 214
    Magnify command 23, **154**
    magnifying glass tool 40
    Reduce/Enlarge commands **153**, 198
    setup file zoom factors 339
    window zoom box 19

DesignWorks™ Simulator 3.1 - Release and Installation Notes

These notes provide supplementary information on DesignWorks Simulator version 3.1, released June 10, 1993.

Thank-you

Thank-you for purchasing the DesignWorks Simulator package from Capilano Computing. We are sure that you will find the package productive and enjoyable to use. Should you have any problems, questions, or comments, please contact us at the number given below.

What you have received

This package contains the 3.1 release of DesignWorks Simulator, and will include the following:

1 Release Notes
5 Diskettes
1 Reference Manual Package
1 License Agreement
1 Serial Number label or sheet §
1 Registration Card *
1 Support Plan Enrollment Card *

* These items will not be included in upgrade packages.

§ If you purchased this package together with the Schematic package, there will be only one Serial Number sheet for both

©1993 Capilano Computing Systems Ltd. All rights reserved.

Note re Version Numbers: In versions of DesignWorks prior to 3.0, the simulator was an integrated function of the package and not available as a separate product. Version 3.1 is actually the first release of the Simulator as a separate product and is so numbered to indicate compatibility with DesignWorks Schematic 3.1.

Installation Notes

See also: Chapter II - Getting Started in the DesignWorks Simulator reference manual.

In order to simplify installation and reduce the number of diskettes required to distribute DesignWorks, the release files are compressed onto a small number of diskettes using the Apple Installer and the StuffIt™ utility.

The Installer allows you to install a complete basic system in one operation, or to choose specific groups of items to be installed.

IMPORTANT: Schematic Version Compatibility

The DesignWorks Simulator requires DesignWorks/Schematic version 3.1.1 or later for correct operation. If you are adding the Simulator to an existing DesignWorks installation, you should install any required Schematic upgrade first.

Installation Options

You may choose to select only a subset of the files included with the Simulator if they are not required for you application. The Installer's "Customize" option allows you to select from among the following groups of items.

Note: Installing the Simulation Tools and Demo Files, etc. items include everything needed for basic Simulator operation, including working through the tutorials.

Simulation Tools	This includes the actual program modules that comprise the Simulator package. They will normally be placed in the Tools folder in your DesignWorks installation. The Simulator is not a stand-alone program and operates only in conjunction with the main DesignWorks package. These items require a total of about 600K of disk space.
Demo Files, etc.	This group includes a number of demonstration designs for use with the Simulator, a basic set of models for use with the tutorials and some utility items. These items require a total of about 1MB of disk space.
74XX Models	This group includes a large number of "external model" files which define the simulation functions for the 74XX and 54XX logic families. These items require a total of about 9 MB of disk space.
4000 Models	This group includes the models for the 4000-series CMOS devices. These items require a total of about 1.2 MB of disk space.

Optional Simulation Tools

The Simulator is made up of a number of different program modules, or "tools" that operate within the DesignWorks environment. Some of these tools are optional and can be removed after installation if they are not required. The following table summarizes the Simulator tools.

Sim	This is the primary digital simulation engine. It is required for all simulation functions.
Timing	The timing diagram display and editing tool. It does not have to be included if timing waveform display is not required.
TestPanel	The test vector entry and execution tool. It does not need to be installed if its services are not required.
SimLoad	This tool associates internal simulation models with parts as they are used and is optional if not needed.

PromPLA	This tool performs data entry and simulation functions for PROM and PLA devices. IMPORTANT: This should be installed even if you are not directly using PROM or PLA devices as many of the simulation models supplied with DesignWorks do use them internally.
RAM	This tool performs data entry and simulation functions for RAM devices.
FromABEL	This tool creates simulation models for devices described using MacABEL or other external means. It can be discarded if you will not be creating PLD or PLA devices.

The Installer places all of the above tools in the Tools folder of your DesignWorks/Schematic installation. Any tools that are not being actively used can be moved temporarily to another folder and they will not be loaded.

Program Installation Procedure

Before starting the installation procedure you should take the following preparatory steps:

1) The Installer requires that it be the only program running on the machine it is installing onto. Before starting the Installer, shut down all other applications.

> Note: If you are using Macintosh System Software version 7.0 or later, the installation will be significantly faster and more problem-free if you restart you machine without extensions. This is done by selecting the Restart item in the Special menu and holding the Shift key on the keyboard depressed until the message "Extensions Off" appears. This disables all file sharing, virus detection and other background processes that can interfere with the installer.

2) If you have any virus detection software on your machine, you may find some interaction with the Installer. It is best to disable any virus software before proceeding.

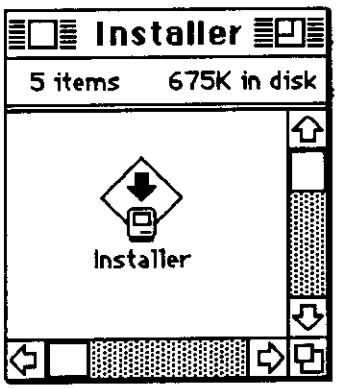
3) Depending on which disk you are installing on, the Installer may require that you disable File Sharing, or at least ensure that no folders are exported from your machine during the installation.

4) The standard installation of DesignWorks and its default libraries requires about 16 megabytes of disk space. Most of this space is taken up by parts

libraries. If you want to install the default system, you will need to ensure that this amount of space is available. You can use the "Customize" option to choose only a basic program installation to get started, or to install additional libraries. In any case, a minimal installation requires at least 5 megabytes.

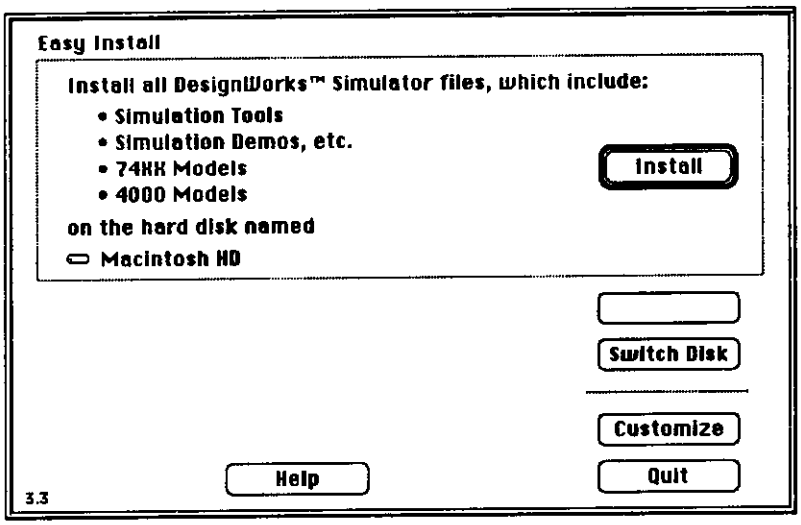
To start the installation:

1) Insert the diskette labeled "Installer". You should see the following files on the diskette:



2) Do not copy any of these files onto your hard disk! Double-click on the file "Installer". This will start up the installer program.

3) The following box will be displayed:



At this point you can choose one of two paths:

- Clicking the Install button will install all the DesignWorks Simulator files on your disk.
- Clicking the Customize button will allow you to choose which items you install.

5) The installer will now copy the files from the installation disks, and then unpack the files onto your disk. It will ask for the diskettes it requires as it proceeds.

6) The next time you start the DesignWorks program, the Simulator tools will appear in the Tools menu.

Serial Number Entry

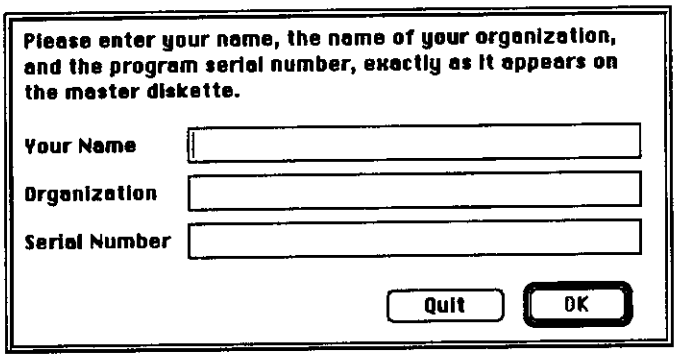
If the Simulator was purchased separately from the original DesignWorks package, you may have to enter the new Serial Number provided in order to enable the Simulator with your copy of DesignWorks. If you purchased the Schematic and Simulator packages together, then simply follow the serial number entry procedure in the DesignWorks/Schematic Release & Installation Notes.

Entering a New Serial Number

To enter a new Serial Number into an existing DesignWorks installation:

- Quit the DesignWorks program if it is already running.
- Start the program by double-clicking on the DesignWorks icon while holding the Option key on the keyboard pressed.

You will see the following serial number entry box:


Please enter your name, the name of your organization, and the program serial number, exactly as it appears on the master diskette.
Your Name
Organization
Serial Number
Quit
OK

Enter the following information:

- Your name and the name of your organization. This information will be displayed each time the program is started.
- The serial number as provided with your master diskette. This is a sequence of 20 letters and numbers. It can be entered either with or without separating spaces. All letters must be upper case.

After this information is entered once, it is kept with the program and you will not need to enter it again.

DesignWorks uses a network copy protection scheme that limits the number of users of a single copy on a network. The serial number contains encoded in it the number of authorized users and validation codes for add-on tools. You can upgrade your copy at any time by contacting Capilano Computing at the phone number given elsewhere in this manual.

Site or Network Licensing

When you purchase a copy of our software you are licensed to store and use it on a single computer at one time, making copies only for backup purposes. We have very attractive multiple unit licensing arrangements for organizations that are interested in using our software on a number of machines on the same network. Please contact us directly for more information.

Copy Protection

For your convenience, the DesignWorks disks are not copy protected in any way, allowing you to copy the program onto your hard disk or backup floppies as desired. This does **not** imply that you are allowed to make copies for use on more than one computer, in fact you are specifically prohibited from doing so by your license agreement. By respecting that agreement, you will help us continue to bring you useful software products at reasonable prices.

Technical Support

If you have any questions, problems, suggestions, congratulations or whatever, you can contact us using any of the following methods:

Phone	(604) 522-6200 (9 - 5 Pacific time, please ask for DesignWorks Technical Support)
Fax	(604) 522-3972
AppleLink	D0816,
InterNet	D0816@applelink.apple.com
Mail	Technical Support, Capilano Computing, 960 Quayside Drive, Suite 406, New Westminster, B. C. CANADA V3M 6G2

Bulletin Board Services

DesignBoard BBS

The DesignBoard BBS is available especially for our customers to access technical support, download updates and technical notes, exchange files and communicate with Capilano personnel.

Access number: (604) 522-4361

Access parameters: 1200/2400/9600 Baud, 8 bits, no parity, 1 stop bit

On your first access you can use basic functions and request validation, or you can phone us in advance and we will create an ID for you.

AppleLink Capilano BBS

Users of the AppleLink network can access the Capilano bulletin board under the Third Party icon for the same services available on our DesignBoard service.

Installer Credits

This program is being distributed using the Apple Installer version 3.3 licensed from Apple Computer, Inc. and *StuffIt* installer technology licensed from Aladdin System. *StuffIt* is a file compression, encryption and file archival utility by Aladdin Systems, 217 E. 86th St., #153, NY, NY, USA.

If you are updating from an older version of DesignWorks

New Features In 3.1

The DesignWorks Simulator version 3.1 contains many important enhancements over the 2.5.x version, including:

• The Simulator now operates on fully hierarchical designs. You can open any subcircuit at any time and probe or modify its internals.

• The Primitive devices have been generalized so that most primitive types can be created in any desired width. E.g. It is now possible to create 17-bit counter or a 128-input AND gate as a single primitive device.

• Pin delays, logic inversion and initial values can be specified individually per pin. This allows path delays through devices to be easily customized.

• Powerful new TestPanel tool allows test vectors to be entered and executed in an interactive, spreadsheet-style format. In addition to straight-forward test vectors, the TestPanel can be used to create a text record of a simulation, to emulate devices or bus interfaces and to monitor simulation results generated by a circuit.

• New "trigger" capability allows automatic detection of specific signal values and time-dependent events such as glitches. Actions including turning the timing display on or off, stopping the simulation and drawing a reference line on the timing diagram can be taken in response to a trigger event.

• The Timing display has many new enhancements, including a timing control panel, bus grouping with hexadecimal value display, text file load and save and many enhanced editing features.

• New simulation model loading and unloading features. The design no longer has to be created and stored with all simulation internal circuits included.

• Input value mapping allows you to specify globally how devices will respond to high impedance and unknown values.

• New display primitive devices include an ASCII display, a 7-segments display and a push-button switch.

• PROM, PLA and RAM devices have been extended. RAM devices can now be loaded from a hex file. PROM and PLA devices are now created from text file input. NOTE: No manual method for PROM or PLA construction is provided in this version.

Updating From 1.x or 2.x

Files created using DesignWorks versions older than 3.0 can be simulated using DesignWorks Simulator version 3.1, but the following points should be noted.

File Format Conversion

DesignWorks 3.x uses a completely new file format and a conversion procedure must be performed on any files created using version 2.5.6 or older. Instructions for this procedure are in the DesignWorks/Converter manual supplied with DesignWorks/Schematic.

Version 3.0 Conversion Errors

The Converter tool supplied with the original DesignWorks 3.0 release did not convert some simulation information in older files correctly. If you wish to simulate these files you must either:

- Reconvert the files from the original 2.x files using the Converter supplied with Schematic version 3.1.1.
- Make a number of corrections to the converted file. These include:
  - Correcting the pin type settings on all "macro" devices in the converted schematic.
  - Verifying the correctness of PLA data. PLA primitive devices marked as Active Low may not be set correctly.
  - Re-initialize RAM data. RAM data was not converted by the original Converter.

Call our Technical Support Department for assistance.

Simulation Setup in Converted Files

Two types of simulation information are not converted when importing older files into version 3.1:

- Timing window setup information: All information about the size and location of the timing window, which signals were displayed and any timing reference lines that were displayed is lost in the conversion process. A new timing window can be created by selecting the Timing item in the Tools menu and then adding traces to it using the Add to Timing command in the Simulation menu.
- Any signal change events that were pending when the file was saved: In particular CLOCK devices will not operate when first opened in version 3.1. The circuit can be restarted by selecting the Clear Simulation item in the Simulation menu. In rare cases, such as a circuit that contains "ring oscillators" or other constructs that require continuous feedback, it may be necessary to manually insert a signal change into the loop using the Signal Probe tool.

Updating from 3.0

DesignWorks 3.0 and 3.1 are fully file compatible and there should be no conversion issues in moving files from version 3.0 to 3.1. Note, however, that the standard symbol libraries supplied with 3.0 and 3.1 contain no simulation information and models will have to be attached to them before they can be simulated.

Refer to Chapter IV - Creating a Schematic for Simulation in the DesignWorks Simulator manual for more information.

This page is intentionally left blank.

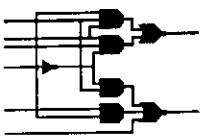
DesignWorks™ Simulator

Version 3.0

for the Apple Macintosh™

Reference Manual

May 17, 1993



Capilano Computing

406 - 960 Quayside Drive
New Westminster, B.C., Canada, V3M 6G2
(604) 522-6200

©1993 Capilano Computing Systems Ltd.

Table of Contents

Chapter I - Introduction 1

Organization of this manual 1
Notes Regarding Copyright 2

Chapter II - Getting Started 3

Installation 3
Installing the Simulation Tools 3
Installing the Device Models 4
Memory Usage 4

Chapter III - Simulation 5

General Information on Simulation 5
Type of Simulation 5
Simulation Memory Usage 6
Time Units 6
Signal Simulation Characteristics 6
Signal States 6
Description of States 7
High Impedance 7
Don't Know 8
Conflict 9
Stuck-at Levels 9
Setting Stuck Levels 10
Clearing Stuck Levels 10
Resolution of Multiple Device Outputs 10
Resistive vs. Forcing Drive 11
Signal Probe Tool 11
Busses 11
Bus Pins 12
Device Simulation Characteristics 12
Device and Pin Delay 12
Primitive Device Delay 12
Sub-Circuit Device Delay 12
Pin Delays 12
Setting the Delay 13
Effect of zero delay 13
Where Delays are Stored 14
Device Storage State 14
Input Signal Values 15
Input Value Mapping 15
Setup and Hold Times 15
Device Pin Types 16
Device Pin Inversion 17

Triggers 18
Simulation Clearing and Initialization 18
The Clear Simulation Operation 18
The Clear Unknowns Operation 19
Setting Initial Values 19
Signal Initial Values 20
Pin Initial Values 20

Chapter IV - Creating a Schematic for Simulation 21

Schematic Simulation Issues 21
Hierarchy Mode 21
Working With Hierarchical Blocks 21
Editing an Open Internal Circuit 21
The Port Interface 22
Parent Device Pin Type 22
Port Connector Pin Type 23
Pin Delays and Inversion 23
Power and Ground Connectors 24
Special Signal Names 0 and 1 24
Simulation Models 25
Primitive Devices on the Schematic 25
Simulation Pseudo-Devices 26
Using External Sub-Circuit Models 26
Using the SimLoad Tool 27
Failure Indications 28
Model Search Folders 28
Selecting Specific Logic Families 29
Batch Loading External Circuit Models 29
Purging External Circuit Models 30
Batch Updating or Reloading Models 30
Model File Naming Conventions 30
Checking Individual External References 31
Customizing Delays for Logic Families 31
Creating External Sub-Circuit Models 31

Chapter V - The Timing Tool 33

The Parts of the Timing Window 33
Displaying Signals in the Timing Diagram 34
Adding a Signal Trace 34
Removing a Signal Trace 34
Repositioning Traces 34
Timing Display Groups 35
Creating a Group Trace 35
Order Within a Group 35
Entering a Group Name 35
The Timing Tool Palette 36

Displaying and Hiding the Tool Palette 36
Tool Palette Time Display 36
Tool Palette Controls 36
Timing Diagram Editing....37
Selecting Data for Copy/Paste Operations....37
Separate Label and Interval Selection 38
Drag Selection....38
Selecting All Traces or All Time 39
Deselecting....40
Summary of Timing Edit Commands 40
The Drawing Tools....41
Chapter VI - The TestPanel Tool....43
General Principles....43
Pop-Up Menus....43
TestPanel Execution Controls....43
Status Display....44
The Data Spreadsheet....44
Data Entry 44
Modes of Operation....45
TestPanel and Hierarchical Designs 45
TestPanel-to-Circuit Association....45
Multiple TestPanels 46
Interaction With Other Simulation Tools....46
Using TestPanel....46
Opening a TestPanel Window 46
Closing TestPanel 47
Editing in the Spreadsheet....47
Editing Text in a Cell....47
Undo 47
Selecting Multiple Cells....47
Moving Between Cells 47
Clipboard Operations....48
Resizing rows and columns 48
TestPanel Controls....48
Step 48
Walk....48
Reset....48
The Test Program....49
Row and Column Usage....49
$Commands....49
Definition Commands....50
Signal List Format 50
The $INPUTS Command....51
The $EXPECTED Command....53
The $OUTPUTS Command....54

The $TIME Command 54
The $WAIT Command 54
The $DELAY Command 55
Control Commands 55
$REPEAT Command 56
$DELAY Command 56
$STOP Command 56
TestPanel Menus 57
File Menu Commands 57
Open Test File 57
Save Test File 57
Set Trace File 57
Save to Design Attr 57
Load From Design Attr 58
Edit Menu Commands 58
Undo 58
Selecting Multiple Cells for Clipboard Operations 58
Exchanging Clipboard Data With The Timing Tool 58
Data Format for Clipboard Operations 59
Cut 59
Copy 59
Paste 59
Clear 60
Select All 60
Fill Down 60
Fill Right 60
Insert Rows 60
Delete Rows 61
Insert Cols 61
Delete Cols 61
Options Menu Commands 61
Link to Circuit 61
Test Options 62
Proceed to Next Step 62
Update Output Values 63
Data Orientation 63
Set Stuck Values 63
Beep at End of Test 63
Clock Width 63
Default Delay 64
Default Header 64
Column Format Options 64
Other Default Header Options 65
Clear Sheet 66
Monitor Mode 66
Copy to Timing 66

Resize Columns 67
Update Display 67
Place and Test Device 67
Save Test to Device 68
TestPanel Examples 68
Testing a Circuit 69
Entering the Test Program 69
Executing the Test Program 70
Dealing With Delays and Setup Times 70
TestPanel as a Pattern Generator 71
Creating Fixed Step Sizes 71
The Test Program 72

Chapter VII - Primitive Devices 73

Gates and Buffers 75
Gate Definition 75
Gate Pin Order 76
Pin Inversions 76
Transmission Gate 77
Three-State Buffer 77
Making Non-Inverting Buffers 78
Resistor 78
Logic Devices 79
Multiplexer 79
Multiplexer Pin Variations 80
Decoder 81
Adder/Incrementer 82
Adder Pin Variations 82
Subtracter/Decrementer 83
D Flip-Flop 83
Flip-Flop Setup and Hold Times 83
Flip-Flop Initialization 84
D Flip-Flop Optional Pins 84
D Latch 84
JK Flip-Flop 85
Register 85
Counter 87
Cascading Multiple Counters 87
Counter Pin Variations 88
Shift Register 89
Clock 89
Creating Synchronized or Offset Clocks 89
Setting Clock Values 90
One Shot 91
Setting One Shot Values 91
I/O Simulation Pseudo-Devices 91

Binary Switch....91
SPST Switch....91
SPDT Switch....92
Logic Probe....92
Hex Keyboard....92
Hex Display....93
SetupHold....93
Unknown Detector....94
RAM Devices....95
Description of the RAM Primitive....95
Don't Know Input Handling....96
RAM Pin Options....96
Creating Custom RAM Devices....96
RAM Size Limits....97
Editing RAM Devices....98

Chapter VIII - Programmable Devices....99

The PROM and PLA Primitive Types....99
PROM Device Format....99
PROM Size Limits....100
PLA Device Format....100
PLA Size Limits....100
Complex Programmable Logic Devices....100
Creating PROMs, PLAs and PLDs for Simulation....101
Creating PROM Devices....101
Generating the Data File....101
The PromPLA Tool....102
Installation....102
Starting PromPLA....102
Editing PROM Devices....103
Reloading PROM Devices....103
Viewing PROM or PLA Contents....104
Creating Programmable Logic Devices....104
The FromABEL Tool....104
Installation....104
Generating the Data File....104
Starting FromABEL....104
The FromABEL Parameters Dialog....105
The FromABEL Synthesis Process....107
Limited or Unsupported PLDs....107
FromABEL Errors....108

Chapter IX - Menu Reference....109

File Menu Commands....109
Open Timing Text....109
Save Timing Text As....110

Print Timing....110
Page Setup....110
Edit Menu Commands....110
Undo....110
Cut....110
Copy....111
Paste....111
Clear....112
Duplicate....112
Point....112
Draw Sig....112
Select All....113
Insert Time....113
Delete Time....113
Simulation Menu Commands....113
Core Simulation Commands....113
Speed....113
Stop....113
Run....113
Other Simulation Speeds....114
Single Step....114
Simulation Params....114
General Delay Box....115
Clock Params Box....117
SetupHold Params Box....118
One Shot Params Box....119
Loading Delays from Attributes....119
Set From Attribute....120
Save to Attribute....120
Triggers....121
Trigger Conditions....121
Trigger Enabling....121
Time Condition Controls....122
Signal Condition Controls....122
Delay Condition Controls....123
Trigger Actions....123
Stick Signals....124
Simulation Scope....125
Clear Simulation....126
Clear Unknowns....126
Signal Probe....126
Probing a Signal....127
Probing a Pin....127
Injecting a Value Using the Probe Tool....128
Timing Window Commands....128
Add to Timing....128

Add as Group 128
Auto Add 129
Find in Timing 129
Timing Options 130
Timing Data Retention 130
Signal Name Display 131
Show/Hide Timing Palette 131
Timing Diagram Pop-Up Menu 131
Get Info 131
Go To Schematic 132
Remove 132
Group 132
Ungroup 132
Collect 132
To Top 132
To Bottom 132

Appendix A - Timing Text Data Format 133

General Description of Format 133
Format Specifics 133
Header Format 134
Single Signal Items 134
Grouped Items 134
Data Line Format 134
Timing Text Example 136

Appendix B - Simulation Attribute Fields 137

Appendix C - Device Pin Types 140

What Pin Types are Used For 140
Pin Types Table 140
Device Pin Type and Simulator Efficiency 141
Bidirectional Pins 142
Output Pins 142
Input Pins 142

Appendix D - Primitive Device Pin Summary 143

Pin Inversion 143
Pin Function Table 143

Credits 147

Index 149

Chapter I - Introduction

The DesignWorks Digital Simulation package consists of a number of tools that work interactively within DesignWorks:

Sim	The digital simulation engine. It is required for all simulation functions.
Timing	The timing diagram display and editing tool. It does not have to be displayed if timing waveform display is not required.
TestPanel	The test vector entry and execution tool. It does not need to be installed if its services are not required.
SimLoad	This tool associates internal simulation models with parts as they are used.
PromPLA	This tool performs data entry and simulation functions for PROM and PLA devices.
RAM	This tool performs data entry and simulation functions for RAM devices.
FromABEL	This tool creates simulation models for devices described using MacABEL.

Organization of this manual

The DesignWorks manual is divided into the following chapters:

Chapter II - Getting Started	Covers installation and initial startup.
Chapter III - Simulation	Details the signal states, time units and other aspects of the simulation. The timing diagram and waveform editing are also discussed.
Chapter IV - Creating a Schematic for Simulation	Discusses signal naming, power and ground, circuit model creation and other issues.
Chapter V - The Timing Diagram	Discusses displaying and editing timing waveforms.
Chapter VI - The TestPanel Tool	Covers the creation and execution of test programs.

Chapter VII - Primitive Devices	Provides information on the simulation characteristics of the primitive device types and the creation and simulation of RAM and connector devices.
Chapter VIII - Programmable Devices	Discusses the creation and simulation of PROM and PLA devices.
Chapter IX - Menu Reference	Provides specific information on the operation of each menu command, except TestPanel menus, which are covered in Chapter VI.
Appendix A - Timing Text Data Format	Describes the text format used for clipboard operations in the Timing tool.
Appendix B - Simulation Attribute Fields	Provides a table of attribute fields used by the various simulation tools.
Appendix C - Device Pin Types	Lists the available pin types and their effect on simulation.
Appendix D - Primitive Device Pin Summary	Provides a summary of pin orders and options for primitive devices.

Notes Regarding Copyright

The DesignWorks Simulator software and manual are copyrighted products. The software license you have purchased entitles you to use the software on a single machine, with copies being made only for backup purposes, unless you have purchase a specific license extension. Any unauthorized copying of the program or documentation is subject to prosecution.

Note regarding trademarks

A number of product trademarks are referred to in this manual. Full credit for these is given in the last section.

Chapter II - Getting Started

This section gives you information on installing and starting up the DesignWorks Simulator.

NOTE: Since the DesignWorks package and the Macintosh system are constantly being upgraded, there may be recent changes and additional information supplied in Release Notes with the package. It is important that you review these notes prior to installation as they may contain information that supersedes that given here.

We will be assuming that you are already familiar with general Macintosh operation, such as copying files, creating folders, etc. If you are not, then work through the introductory sections of the Macintosh user's guide.

IMPORTANT: Make a backup copy

Before proceeding, make a backup copy and a working copy of your DesignWorks disks, then put the originals away for safekeeping.

Installation

Installing the Simulation Tools

The following table summarizes the code modules making up the Simulator.

Tool	Status	Minimum Memory Usage
Sim	Required	100K
Timing	Optional	115K
TestPanel	Optional	80K
FromABEL	Optional	78K
RAM	Optional	45K
PromPLA	Optional	40K
SimLoad	Optional	33K

In order to be loaded when DesignWorks starts up, these tools must be in the same folder with the DesignWorks program itself, or in the folder specified in the TOOLSFOLDER statement in the DesignWorks setup file, normally "Tools".

Installing the Device Models

Simulation models for the standard digital library components are supplied in the form of "external sub-circuit models", described elsewhere in this manual. These should normally be located in a folder called Models inside the DesignWorks folder.

See the Installation Notes supplied with the package for specific installation instructions.

Memory Usage

Each tool occupies a certain amount of memory space regardless of whether it is being used or not. If you are operating on a system with a minimal amount of memory, it may be desirable to keep unused tools in another folder where it will not be found and loaded by DesignWorks . When it is actually needed, it can be moved temporarily into the Tools folder.

While operating, the Simulator tools can consume large amounts of memory depending upon the size of the design and the amount of data being displayed.

Chapter III - Simulation

This chapter provides more detailed information on the simulation capabilities of DesignWorks.

General Information on Simulation

DesignWorks has the ability to perform a realistic simulation of any digital circuit. Obviously, though, any simulation of any system must be limited in detail and must make certain assumptions. In particular, when simulating digital circuits, it must be understood that real circuits are never completely "digital" in nature, and in fact have many "analog" properties which affect how they operate.

DesignWorks is primarily intended to assist with the logical design of a circuit, and does not take into account factors such as line loading, power supply noise, rise and fall times, output drive, etc. As more of these factors are taken into account, the simulation becomes slower and less interactive, which defeats the purpose for which DesignWorks was created. However, for users with programming capability, the MEDA interface allows programmed simulation models of arbitrary complexity to be added to DesignWorks. These external modules can take into account any desired device characteristics.

Type of Simulation

DesignWorks performs a discrete simulation of the signal changes in a logic circuit, meaning that signal levels and time change only in steps, rather than continuously. The program does not attempt to analyze your circuit, but simply tracks signal level changes through the devices. Thus, circuits with feedback loops or other delay-dependent features will be simulated correctly as long as they don't rely on particular analog characteristics of devices.

The simulation is "event driven", an event being a change in level of a signal. Each time an event occurs, a list is made of all the devices whose inputs are affected by that event. Any other events occurring at the same time are similarly evaluated, and affected devices added to the list. A type-specific routine is then called for each device on the change list in order to determine what output changes are going to occur. These changes are added to the event list, their time of occurrence depending upon the device delay. No computation is performed for times when no event occurs, so that device delay settings and clock values have no effect on how fast the simulation is performed.

DesignWorks performs strictly a digital simulation. It does not take into account factors such as fan-out (i.e. the number of inputs connected to a given output), line length (capacitance), asymmetrical output drive, etc., except in as much as these affect delay time.

Simulation Memory Usage

When a circuit is opened or created by DesignWorks, the circuit data is retained completely in the memory of your machine. Since the total memory available is fixed (until you buy your next memory expansion!), this places some limits on circuit size and simulation.

Each time a signal changes state, an "event" record is created in memory. If the signal is not being displayed in the timing diagram, then this record is deallocated again after the signal change has occurred. If the signal is being displayed, then the record is retained in memory until that change has scrolled off the left-hand side of the timing diagram. As a result, the memory used by event records will increase when the number of displayed signals is increased the resolution of the timing display is decreased or the "retain time" setting is increased.

Time Units

DesignWorks uses 32-bit signed integer arithmetic to calculate all time values used in the simulation. It is usually convenient to think of these values as being in nanoseconds, but the actual interpretation is left up to the user.

The simulation will stop if any time value approaches the 32-bit integer limit.

Signal Simulation Characteristics

Signal States

DesignWorks uses thirteen different device output states in order to track conditions within your circuit. These states can be broken into three groups, as follows.

Forcing States (denoted by suffix .F):

LOW.F

HIGH.F

DONT01.F

DONT0Z.F

DONT1Z.F

CONF.F

Resistive States (denoted by suffix .R):

LOW.R

HIGH.R

DONT01.R

DONT0Z.R

DONT1Z.R

CONF.R

High Impedance:

HIGHZ

Note that the Forcing/Resistive distinction is used only to resolve conflicts between multiple outputs connected to the same signal. The final value stored or displayed for a given signal line can only be one of five possibilities:

LOW

HIGH

DONT

CONF

HIGHZ

Description of States

The High and Low states are the normal ones expected in a binary circuit, but are not sufficient to realistically simulate circuit operation, so the High Impedance, Don't Know and Conflict states are added. There will always be some cases where the simulation will not correctly mimic what would appear in a real circuit, and some of these cases are discussed below. In particular, if a circuit takes advantage of some analog property of a specific device, such as inputs that float high, known state at power-up, input hysteresis, etc., it will in general not simulate correctly.

High Impedance

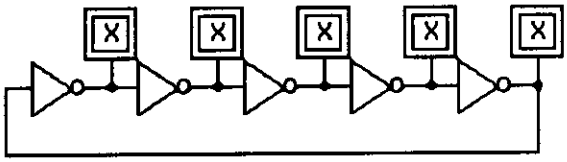
This state ("Z" on a logic probe) is used for cases when no device output is driving a given signal line. This may occur for an unconnected input, or for a disabled "three-state" or "open-collector" type device. If a device input is

in the High Impedance state, it is treated as unknown for the purposes of simulation, even though in a real circuit the device may assume a high or low state, depending on the circuit technology used.

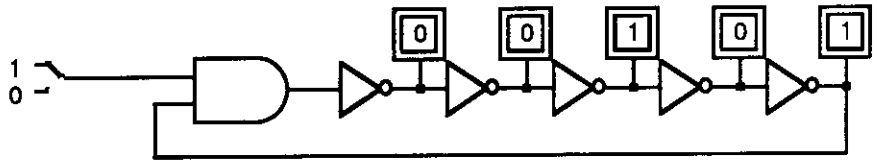
Don't Know

The "Don't Know" state ("X" on a logic probe) results when the simulator cannot determine the output of a device. This may occur, for example, when an input is unconnected or the output from a previous device is unknown. The Don't Know signal will be propagated through the circuit, showing the potential effects of that condition.

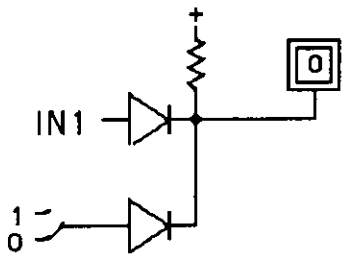
The Don't Know state is used in DesignWorks in cases where the actual result in a real circuit would depend in the circuit technology used, on random chance, or on analog properties of the device not predictable using a strictly digital simulation. For example, if the following ring oscillator circuit is created in DesignWorks, all signals will be permanently unknown, since each depends on the previous one, which is also unknown. In actual hardware, this circuit may oscillate, or may settle into an intermediate logic level, which would not be defined in a digital circuit.



For the purposes of simulation, all circuits must have some provision for initialization to a known state. In most cases, circuits can be initialized using the Clear Unknowns command or by setting the initial value attribute, both described elsewhere in the manual. Alternatively, circuitry can be added to allow a reset to be done, as in the following modification to the ring oscillator:



A problem arises in simulating circuits with multiple open collector devices, such as a bus line, illustrated below:

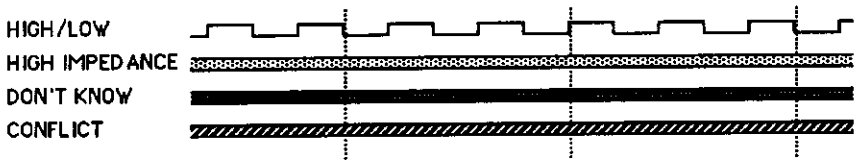


In this circuit the upper device has an unconnected input at IN1 and therefore outputs a "Don't Know" value. The lower device has a low input and therefore outputs a low value. In order to correctly resolve this situation the simulator needs to distinguish between a "Don't Know" output from a normal "totem-pole" type output and a "Don't Know" from an open collector, open-drain, or other single-drive output. In this case the upper device will produce a DONT0Z output which resolves correctly to a LOW on the output regardless of the state of IN1, using the rules described above.

Conflict

The "Conflict" state ("C" on a logic probe) results when two device outputs are connected and are of different or unknown state taking into account the rules described above.

The following timing diagram shows how the various signal states are displayed.



Stuck-at Levels

The DesignWorks simulator implements "stuck-at" levels to assist in setting initial simulation states, testing for faults, etc. When a signal is in a "stuck" state, it will not change state, regardless of changes in devices driving the line.

When the stuck status is set, the signal will retain the value it had at the time until some user action forces a change. When the stuck status is

removed, the signal will return to the value determined by the devices driving the line.

Setting Stuck Levels

A signal can be placed in a Stuck-High or Stuck-Low state by any of the following means:

- applying the name "0" or "1" to the signal.
- typing "H" or "L" while viewing the signal value with the signal probe tool.
- using the Stick High or Stick Low buttons in the Stick Signals command.
- setting a "stuck" option in the TestPanel or other external simulation tool.

Each of these methods is described in more detail in the relevant sections of this manual.

Clearing Stuck Levels

The stuck status can only be cleared by one of the following user actions:

- typing the spacebar while viewing the signal using the signal probe tool.
- clearing the "stuck" state using the TestPanel or other external simulation tool.
- clearing the "stuck" switch in the Stick Signals command

Resolution of Multiple Device Outputs

The DONT0Z and DONT1Z values are used primarily to handle cases of open collector or open emitter devices with unknown inputs (see more information below). Most other types of devices produce the DONT01 output when a value cannot be calculated.

In cases where two or more device outputs are connected together and each one drives the line with a different value, the following rules are used to resolve the actual value on the line.

- the forcing/resistive distinction is only used to resolve outputs from multiple devices. The final value used for display and simulation purposes is one of the forcing values or HIGHZ.
- a forcing drive always overrides a resistive drive or HIGHZ (i.e. the signal takes on the value of the forcing drive, ignoring all resistive drives and HIGHZs).
- a resistive drive always overrides HIGHZ.

- DONT0Z.F and LOW.F produce LOW.
- DONT1Z.F and HIGH.F produce HIGH.
- any other combination of conflicting forcing drives produces CONF.
- DONT0Z.R and LOW.R produce LOW.
- DONT1Z.R and HIGH.R produce HIGH.
- any other combination of conflicting resistive drives produces CONF.

Resistive vs. Forcing Drive

All primitive devices in DesignWorks output a forcing drive level except for the Resistor primitive device. The function of the Resistor device is to convert a forcing drive on one side into a resistive drive on the other. This can be used to modify the output of any existing device type by placing a resistor in series with it. Note that DesignWorks does not model analog properties of devices, so the resistor does not have a resistance value in the analog sense. In particular, there is no interaction between resistor and capacitor symbols to produce delay in lines. The delay effect can be simulated by setting a delay value for the resistor.

Signal Probe Tool

The Signal Probe tool allows you to interactively examine and change values on individual signals and pins in the circuit diagram. When the probe tip is clicked and held on a signal line or pin, the cursor will show the current value on the signal or pin and track changes that occur as the simulation progresses.

See Chapter IX - Menu Reference for more information on the Signal Probe tool.

Busses

Busses (i.e. groups of signals represented by a single line on the schematic) have no particular significance to the simulator. The value of a bus is completely determined by the values of the individual signals it contains. The simulator performs no operations on the bus itself.

NOTE: You can display a bus in the timing diagram using the Add To Timing command, but this is really equivalent to displaying all the internal signals individually and then grouping them.

Bus Pins

Bus pins, like busses, have no particular significance to the simulator. The value of a bus is completely determined by the values of the individual pins it contains. The simulator performs no operations on the bus pin itself. Bus pins are not supported on primitive device types.

Device Simulation Characteristics

Device and Pin Delay

Primitive Device Delay

Primitive devices (i.e. those with a program-defined simulation model) have a single delay value which can be set to any integer value from 0 to 32767. This delay is applied when any input change causes any output change. In addition, a pin delay in the range 0 to 32767 can be set on any input or output pin. Pin delays can be used to set arbitrary path delays through the device. See more information on pin delays below.

The initial delay value is set to 1 when the device is created, but this can be changed later using the Simulation Params command in the Simulation menu. This delay applies whenever any input change causes an output change. There is no provision in the built-in simulation models for different delay values on low-to-high and high-to-low transitions. The Clock and I/O devices have no delay characteristic. See the notes below on delay in hierarchical designs.

Sub-Circuit Device Delay

Sub-circuit devices inherit their delay characteristics from their internal circuit and have no "device delay" characteristic of their own. Simulation Params cannot be directly used on a sub-circuit device, although pin delays can be set separately on each instance of a sub-circuit device to customize path delays.

Pin Delays

Any input or output pin on any device (including port connectors and sub-circuit devices) can have a pin delay associated with it. Pin delays normally default to 0 time units, but can be in the range 0 to 32767.

A pin delay acts like a "buffer" device with the given delay inserted in line with the pin. On an input pin, the device simulation model will not see a change in signal value until after the pin delay has elapsed. On an output pin, the pin delay is added to the overall device delay for any changes scheduled on that pin.

Setting the Delay

Set the delay for a device by first selecting the device (i.e. by clicking the mouse button with the arrow cursor positioned in the device symbol), and then choosing Simulation Params from the Simulation menu.

A dialog box will appear allowing you to increase or decrease the delay value by clicking one of two buttons. The minimum delay value is 0 and the maximum is 32767. When the delay setting for a sub-circuit device is changed, the delays for all internal devices are changed by the same amount.

Effect of zero delay

A delay value of zero is permitted in a DesignWorks device, but this setting should be used only with an understanding of how the simulation is implemented as it can result in unexpected side-effects.

Note that on a given pass through the simulation routine, all the events on the list which occur at the current time are scanned and then the new outputs for all affected devices are calculated. If any of these devices has a zero delay setting then this will result in more changes being placed on the event list at the current time. However, all these changes emerging from zero-delay devices will not be evaluated until the next pass through the simulator. This is done to allow for user interaction with the simulation.

If you step interactively through a circuit with zero-delay elements, you will see all these value changes updated on the screen, even though "simulation time" does not advance. If a signal changes value then reverts to its original state within the same time step, this will be displayed as a zero-width spike on the timing diagram.

If a zero-delay feedback loop exists in a circuit, the signal changes will be simulated and any probes on the diagram will be updated at each pass through the simulator. However the events at the head of the list will always have the same time value associated with them and the simulated time will never advance. This will result in the timing window updating stopping until some delay is inserted in the loop.

Where Delays are Stored

Delays are stored as decimal integers in text form in attribute fields associated with each device or pin. For devices, the delay field is called "Delay.Dev", for pins it is "Delay.Pin" An empty or invalid string will be interpreted as the default value, usually 1 for devices and 0 for pins.

Some special-purpose devices, such as the Clock, One Shot and SetupHold primitive devices take two delay characteristics. In this case, two integers separated by a comma should appear in the Delay.Dev field.

More information on this is given in the information section on each of these primitive types in Chapter VII - Primitive Devices.

A number of additional fields are pre-defined in DesignWorks for storage of alternate delay values. These fields are not used directly by the simulator, but are only provided as a convenient place to store these values. The Save to Attribute and Set from Attribute functions in the Simulation Params command can be used to copy values to or from alternate delay fields.

The following alternate fields are pre-defined.

Field Name	Function
Delay.Dev.Typ Delay.Pin.Typ	Typical device/pin delay
Delay.Dev.Min Delay.Pin.Min	Minimum device/pin delay
Delay.Dev.Max Delay.Pin.Min	Maximum device/pin delay

Device Storage State

In DesignWorks, primitive storage devices (e.g. flip-flops, counters and registers) do not store their current state internally. The device state is completely determined by the values on the signals attached to the output pins. Thus, the following factors will affect the operation of these devices:

- Conflicting or overriding values on the output signals (e.g. a stuck state) will override the last device state calculated by the model.
- Device and pin delays will influence the calculation of a new device state. E.g. if the period of a clock applied to a counter is less than the total delay through it, an erroneous count sequence will result.

If desired, this behaviour can be modified by placing the primitive devices in a sub-circuit device and setting appropriate pin types and delays on the parent device to "buffer" the outputs.

NOTE: These comments do not apply to RAM or bidirectional switch primitives, both of which store internal state information independent of the values of the attached signals.

See the section "Working With Hierarchical Blocks" in Chapter IV - Creating a Schematic for Simulation for more information.

Input Signal Values

Unless an alternate input value mapping is specified (see below), for all device types except switches, signal values "High-impedance" and "Conflict" are treated as "Don't Know" when applied to a device input. When a device is first created, all input signals take the "High Impedance" state and outputs are set depending on their type, normally to the "Don't Know" state. Thus an unused input pin will appear as an unknown input to a device, which may affect its output level.

As with real circuits, all unused inputs should be connected to a high or low level as appropriate. This can be done by naming the pin signal either "0" or "1", using a power or ground symbol, or by using a pullup resistor to set a high level. See more information on logic states in other parts of this chapter.

Input Value Mapping

You can specify an alternate input value mapping for a design by placing a specially-formatted string in the Sim.InputMap attribute field for the design. This is intended primarily to model specific logic families that have a known response to high-impedance inputs. The mapping applies only to primitive devices and is global to the design that it is specified in. Note that this mapping does not occur at inputs to sub-circuit devices but does affect the primitives that they contain.

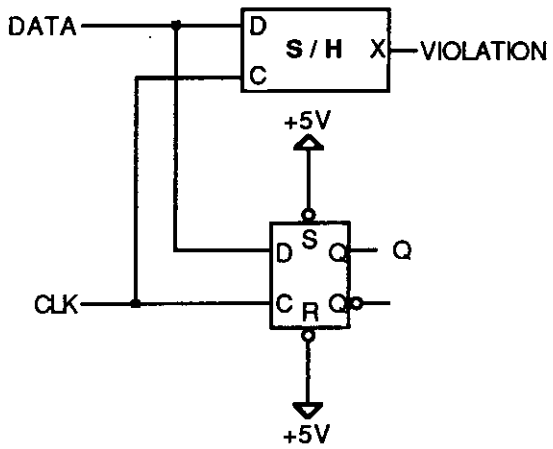
See Appendix B - Simulation Attribute Fields for information on the format of the Sim.InputMap field.

Setup and Hold Times

All standard device types having an edge-triggered clock, such as the D- and JK-flip-flops, register and shift register, have an effective setup time of 1

unit and a hold time of zero units. That is, if the data and clock inputs change simultaneously, the old value of the data input will be used.

Setup and hold times can be checked by attaching a "SetupHold" primitive device to the inputs of the clocked device to be checked, as follows:



The SetupHold device puts out a highZ value until a setup or hold violation occurs when it switches to a don't know state. Thus, the output of the SetupHold device can be paralleled with the flip-flop so that the output line will enter a conflict state when an error occurs.

See more information on the SetupHold device in Chapter VII - Primitive Devices

Device Pin Types

Every device pin has a characteristic known as its *pin type*, e.g. input or output. The pin type is set when the part entry in the library is created and cannot be changed for individual device pins on the schematic. Correct pin type settings are crucial to correct and efficient operation of the simulator.

The pin type is used by the simulator to determine the direction of signal flow and which output values are allowable on a given output pin.

For detailed information on the available pin types and how they affect the simulation see Appendix C - Device Pin Types. For procedures for setting pin types when creating a symbol see Chapter X - Device Symbol Editing in the DesignWorks/Schematic Reference Manual.

Device Pin Inversion

The logic of any pin on any device can be inverted by placing a non-empty value in the Invert.Pin attribute field of the pin. When this is done any value passing into or out from that pin will be inverted. This applies to primitive types as well as sub-circuit devices. The following table summarizes the level mappings that occur.

External Signal Value	Internal Signal Value
LOW.H	HIGH.H
LOW.L	HIGH.L
HIGH.H	LOW.H
HIGH.L	LOW.L
All others	Unchanged

NOTES:

1) The logical inversion of the pin is completely independent of the graphical representation of the pin. I.e. using the "inverted pin" graphic in the DevEditor does not invert the pin logic in the simulator. You must set the Invert.Pin field to have this effect.

2) Although pin inversion can be specified independently for each device on the schematic we do not recommend modifying these settings after a device has been placed on the diagram. This can create the confusing situation of two devices with the same name and symbol but different logical characteristics.

See also:

- *Appendix D - Primitive Device Pin Summary for more information on pin usage and pin inversion.*
- *Chapter VII - Primitive Devices for information on how pin inversion can be used with specific primitive types.*
- *Chapter IV - Creating a Schematic for Simulation for information on pin inversion in sub-circuit blocks.*
- *Chapter X - Device Symbol Editing in the DesignWorks/Schematic Reference Manual for procedures for setting pin attributes when creating a symbol.*

Triggers

The DesignWorks Simulator has a powerful trigger capability analogous to a "word recognizer" on a logic analyzer. Any number of triggers can be set to perform various actions when certain time and signal value conditions are met. These actions can be:

- Drawing reference lines on the timing diagram.
- Stopping the simulator.
- Enabling or disabling the timing display.
- Generating a beep in the computers loudspeaker.
- Enable checking for another trigger.

Any number of triggers can be set up with different activation conditions. For display purposes, triggers are named T1, T2, T3, etc. To assist in setting up complex time-related conditions, trigger T_N can be used to enable T_{N+1}.

See Chapter IX - Menu Reference for specific details on setting up triggers.

Simulation Clearing and Initialization

The DesignWorks Simulator provides a number of mechanisms to assist in setting initial values and restarting a simulation.

The Clear Simulation Operation

The Clear Simulation operation is invoked by either selecting the Clear Simulation command in the Simulation menu or by clicking on the Restart button on the timing tool palette. This operation performs the following steps:

- Other tools (such as Timing and TestPanel) are notified and perform their own processing.
- All signal change events on the queue are disposed of, whether pending or historical.
- Any clocks in the design are re-initialized.
- If any signal or pin initial values are specified, they are set up. See below for information on setting initial values.
- All devices are queued for immediate reevaluation.

The Clear Unknowns Operation

The Clear Unknowns operation is a heuristic procedure which attempts to remove "Don't Know" signal values from a design. This can be used to find an initial state when a design is first simulated or after any edit operations that result in unknown values. This operation is invoked by either selecting the Clear Unknowns command in the Simulation menu or clicking on the Clear X button on the timing tool palette.

The Clear Unknowns operation performs the following steps, stopping as soon as all unknown states are removed from the design:

- Any pending signal change that would result in an unknown state is removed from the queue.
- Any primitive type with storage capability (i.e. flip-flop, register or counter) that has a Don't Know output value is cleared, either to its specified initial value (if any) or to zero.
- The input mapping for the design is temporarily set so that all unknown input values map to zero.
- A single device that currently has an unknown output state is randomly selected and queued for reevaluation. The simulator is cycled repeatedly as long as the number of unknown states in the design decreases. This step is then repeated until the number of unknowns ceases to diminish.
- The input mapping for the design is restored to its original state.

If this operation does not clear the design to an appropriate state, refer to the other techniques discussed below.

NOTE: Design's with "hard" unknowns, such as unconnected inputs or conflicting outputs will not be successfully cleared by this procedure. All device inputs should be specified to a known value if not driven by other devices.

Setting Initial Values

Initial values can be specified for signals and pins which will be applied by the Clear Simulation and Clear Unknowns operations, as described in the preceding sections.

For both object types, the initial value is entered into an attribute field, either Initial.Sig or Initial.Pin. The allowable values consist of a single character chosen from the following table.

Character	Value
0	LOW
1	HIGH
Z	HIGHZ
X	DONT01

All other values will be ignored.

NOTES:

1) It is left completely to the user to decide if the specified initial values make sense. E.g. No checking is done to determine if a given device output value is the reasonable result of the device's current inputs.

2) Devices do not have initial value settings since their values are completely determined by the state of their output pins. See Pin Initial Values below.

Signal Initial Values

The initial value for a signal is stored in the Initial.Sig attribute field using the format described in the previous section. When a Clear Simulation operation is invoked, the initial value specified is placed on the signal without regard for the current output levels of devices driving the signal. The given value will stay on the signal until some device driving the signal changes state or some other user action changes it.

NOTE: If a pin initial value is specified for any output pin driving the signal, the signal value will be overridden.

Pin Initial Values

The initial value for a pin is stored in the Initial.Pin attribute field using the format described earlier. Initial values can only be specified for output or bidirectional pins and will be ignored on input pins.

When a Clear Simulation operation is invoked, the initial value specified is placed on the pin without regard for the current inputs affecting the device. The given value will stay on the pin until the device model schedules a state change or some other user action changes it.

Chapter IV - Creating a Schematic for Simulation

Schematic Simulation Issues

Hierarchy Mode

The DesignWorks Schematic tool supports three hierarchy modes: Flat, Physical and Pure. The following table summarizes these modes in terms of their significance for simulation.

Flat	Despite its name, Flat mode does allow devices to have internal circuits for simulation purposes, although it will discourage access to them. Flat mode also changes the way device names are assigned and reports are generated, but these issues do not affect the simulator.
Physical	This mode allows unrestricted access to internal circuits for simulation purposes.
Pure	Simulation is not supported in Pure mode. This is because only a single definition copy of device type data is kept in memory, regardless of the number of times the device is used. There is nowhere to store the simulation states associated with each instance.

Working With Hierarchical Blocks

The simulator does not impose any new rules on working with hierarchical blocks, but there are some effects of editing a design with active simulation that should be noted.

See also Chapter VII - Hierarchical Design in the DesignWorks/Schematic Reference Manual.

Editing an Open Internal Circuit

A number of issues arise if you have used the same device type or hierarchical block multiple times in a design and you open one copy for editing (i.e. using the Push Into command or by double-clicking on the device). You should note the following points:

• The Schematic tool creates a separate, temporary type definition for the open device when it is opened. Any simulation values that you view or change, or any circuit changes made will apply only to that one device instance while it remains open.

• When you close an open internal circuit, the action taken depends on edits that have taken place. If you have made no edits or changed only instance data (i.e. simulation values or attributes marked as "keep with instance"), then the other blocks of the same type will not be affected by the close. If you have changed any definition data (i.e. any graphical or structural change to the circuit or any definition attribute change) then you will be prompted for the action to take. If you click the Update button, all instance data (i.e. signal values, etc.) from other blocks of the same type will be lost. It will be completely replaced by the values from the edited block.

The Port Interface

The connection between a pin on a parent device symbol and the corresponding signal in the internal circuit is quite complex from a simulation standpoint. In order for this connection to act like a "hard wire" between the two levels, the following conditions must be met:

• the *pin type* on the parent device symbol must be "bidirectional".
• the *pin type* of the corresponding port connector in the internal circuit must be "bidirectional".
• the *pin delays* on both the pin on the parent device and the pin on the port connector must be zero.
• no *pin inversion* must be specified either on the parent device pin or the port connector pin.

Any other combination of settings will result in some degree of isolation or "buffering" between the two levels. I.e. The observed signal value on the signal in the internal circuit may be different from that on the parent pin.

NOTE: When a symbol is created in the DevEditor, all pins default to type "input", i.e. they will not drive any attached signal. If you are creating a hierarchical block symbol for simulation purposes, the pin types must be set to appropriate values.

The effects on these various settings are summarized in the following sections.

Parent Device Pin Type

Any signal value driven out of a parent pin by an internal circuit may be translated according to the pin type on the parent device. These effects are summarized in the following table.

Pin Type	Effect
Input	This will prevent that pin from ever driving the attached signal, regardless of drives in the internal circuit.
Output/Three-state	This will pass the sum of the internal drives up to the parent pin without any translation. Signal value changes on the signal attached to the parent pin will not be passed to the internal circuit.
Open collector/Open emitter	Any drive level from the internal circuit will be translated according the capability of the pin type. See Appendix C - Device Pin Types for more details.
Bidirectional	All changes on the internal signal are passed to the parent pin and vice versa.
Other types	Other types, such as Tied High and Tied Low are not recommended.

NOTE: Although it may be tempting to set all pins to "bidirectional", this is not recommended since it significantly increases simulation overhead and increases the difficulty of isolating circuit drive problems.

Port Connector Pin Type

The pin type on the port connector is also used to translate the value of any incoming signal changes, in a manner similar to the parent pin type. Normally, the pin type setting on a port connector should complement the setting of the parent pin as follows:

Parent Pin Type	Port Connector Name	Port Connector Pin Type
Input	Port Conn In	Output
Bidirectional	Port Conn Bidir	Bidirectional
All others	Port Conn Out	Input

Other settings on the port connector pin are not recommended.

Pin Delays and Inversion

The normal pin delay and inversion settings can be applied to the port interface. A non-null value in the Invert.Pin attribute will cause any signal values passing in either direction to be inverted. An integer value in the

Delay.Pin attribute will cause the specified delay to be inserted in line with level changes passing in either direction.

NOTES:

1) It is recommended that pin delay and inversion settings be applied only to the pin on the parent device and not to the port connector in the internal circuit. Attribute settings on the port connector are more difficult to verify and edit since the port connector is a "pseudo-device" and some schematic editing operations will be disabled.

2) Changes made in Invert.Pin and Delay.Pin after a device has been placed on the schematic, will affect only that one device instance. Default values can be set in these attribute fields when the symbol is created in DevEditor.

Power and Ground Connectors

Power and Ground connector symbols do not have any inherent simulation signal drive unless their pin type has been set to "Tied High" or "Tied Low", as appropriate. Most of the positive-supply symbols provided with DesignWorks 3.1 have "tied high" settings, while others will be "tied low". The symbols provided with older DesignWorks releases may not have any drive setting, resulting in a high impedance level on these signals. This can be remedied by either:

- Replacing any one or all the ground or power symbols with one with the appropriate setting.
- Forcing a Stuck High or Stuck Low level onto the signal using the signal probe tool, TestPanel tool or other external means. Note that because all like-named ground or power segments are logically connected, this only needs to be done on any one segment.

Special Signal Names 0 and 1

The signal names "0" and "1" are recognized by the simulator as special. If any signal is named "0", it will be given a Stuck Low value. If "1" is found, it will be given a Stuck High value. These values can be cleared or changed using the signal probe, if desired.

See the Signal Probe command in Chapter IX - Menu Reference for more information.

Simulation Models

In order to completely simulate a design, every symbol must have an associated simulation model. In DesignWorks, simulation models can take one of the following forms:

- Primitive Devices - These types have "hard-wired" program code to evaluate input and output changes. They include the gates, flip-flops and other devices described in Chapter VII - Primitive Devices, as well as the user-definable PROM and PLA primitives.
- Sub-circuit Devices - The simulation function of a sub-circuit device is completely determined by its internal circuit (except for the addition of pin delays and inversion). The definition of a device sub-circuit can be stored with the part in a library (referred to as an "internal sub-circuit") or can be stored in a separate design file on disk (an "external sub-circuit"). The sub-circuit itself can contain any combination of primitive devices or other sub-circuits (except itself, of course!) nested to any desired depth.
- MEDA Code Models - It is possible to associate an external block of program code with a device for simulation purposes. Whenever any input change or editing event affects the device, the program code will be executed to generate the new output values. This program code can take any desired inputs into account, including disk files, user input, real hardware, etc. Creation of these models requires the optional MEDA Developer's Kit and is beyond the scope of this manual.

Whenever any device type is to be simulated, all information about the device must be loaded into memory. Unless you explicitly purge internal circuits or code models from the design, they will become a permanent part of the design and will be saved with the file.

Primitive Devices on the Schematic

The primitive devices provided in the "Primitive xxx" libraries can be used at any time as part of a schematic, whether or not the simulator is installed. However, these libraries are not intended to match any real logic families and do not have any part name, pin number or packaging information associated with them.

See Chapter VII - Primitive Devices for more information on creating and using primitive types.

Simulation Pseudo-Devices

The simulation pseudo-devices (i.e. those in the Primitive I/O library) are handled specially by the Schematic tool. In general, you cannot modify the symbol, pin types or other characteristics of these devices. In addition, they are treated differently from normal device symbols in the following ways:

- By default, these devices are flagged "omit from report", meaning that they will not appear in any netlist or bill of materials produced by the Report tool. This setting can be changed using the Schematic tool's Get Info command.
- These symbols will not be assigned a name when placed on a schematic. A name can be manually assigned, if desired.
- The switch and keyboard types respond to a normal mouse click by changing state, rather than being selected. To select one of these device, hold the Shift key pressed while clicking on it.

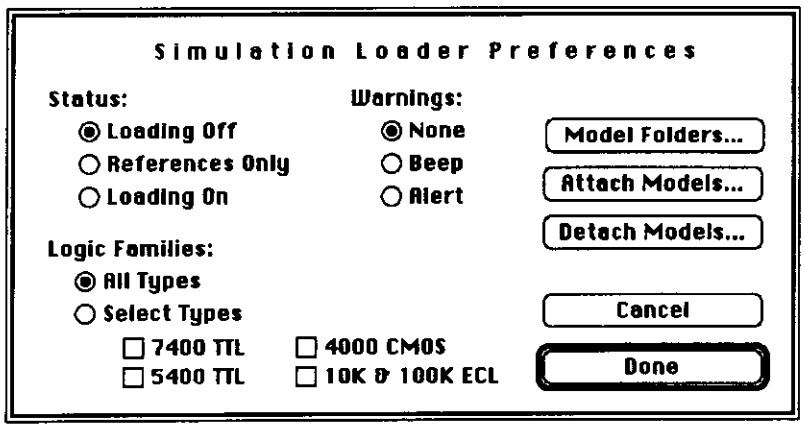
Using External Sub-Circuit Models

Most of the models for industry-standard device types (e.g. 74XX, 4000, etc.) are provided as external sub-circuits. These models are simply design files containing an equivalent circuit for the device. The circuits are designed to match with the symbol libraries provided with the Schematic package.

External sub-circuit models can be loaded either interactively, i.e. as each part is used, or in batch mode after the design is created. Both of these functions are provided by the SimLoad tool.

Using the SimLoad Tool

SimLoad defaults to the inactive state, i.e. no loading occurs. It is activated by selecting its name in the Tools menu. It will then display the following box:



Two levels of loading action are available:

- References Only - This method can be used to check for the existence of the model file without actually loading it. In this case, each time a new part is used, SimLoad attempts to locate a model for it and sets the external circuit reference attributes to the location of the file. The file itself is not loaded, i.e. the device will not have a sub-circuit. The sub-circuit can be loaded later using the Attach External Sub-Circuits command in the Sub-Circuit sub-menu in the Schematic tool's Options menu.
- Loading On - In this mode, SimLoad locates the external model file and immediately loads and attaches it to the part. The external reference attributes are also set, allowing for later updating if the model file is changed.

NOTE: The SimLoad action in both the above cases is the same. In "Loading On" mode, the SimLoad tool commands the Schematic tool to enable its "Auto-Load External Sub-Circuits" mode. This mode can also be controlled directly in the Sub-Circuit sub-menu in the Schematic tool.

Failure Indications

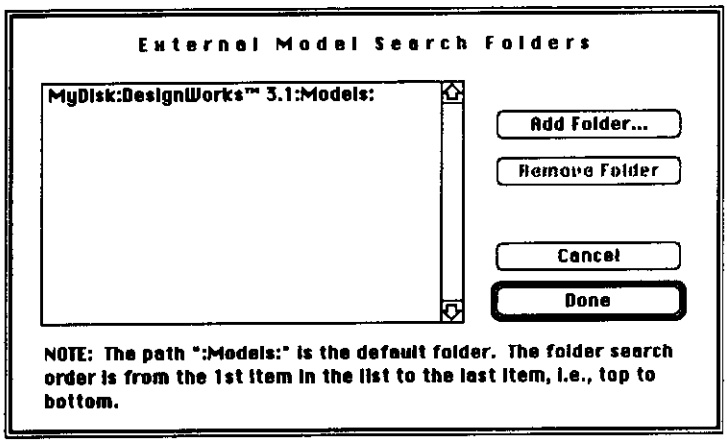
The "Warnings" selections in the SimLoad box allow you to choose the level of warning you receive if a model is not located.

None	No warning is issued.
Beep	A standard beep tone will be issued each time a new device type is used for which a model could not be located.
Alert	A warning box will be displayed whenever a model is not located.

It is possible to determine at any time which devices have an external model associated with them by viewing the contents of the ExtCctName attribute field. This can be done using the Browser tool (select the ExtCctName field in the Secondary field list), or using the Report tool. A special report format file called "External Circuit Form" is provided with the simulator for this purpose.

Model Search Folders

By default, the SimLoad tool assumes that model files are located in a folder called Models in the same folder as the DesignWorks program itself. It is possible to place additional folders on the search list by clicking on the Model Folders button in the SimLoad box. This will display the following box:


External Model Search Folders
MyDisk:DesignWorks™ 3.1:Models:
Add Folder...
Remove Folder
Cancel
Done
NOTE: The path ":Models:" is the default folder. The folder search order is from the 1st item in the list to the last item, i.e., top to bottom.

To add a new folder, click on the Add Folder button. This will display a standard file selection box. In this file box, the "Directory" button adds the directory selected at left to the search list. The "Select Current Directory" button adds the directory containing the current items to the list, i.e. the

directory whose name shows in the file level pop-up menu at the top of the box.

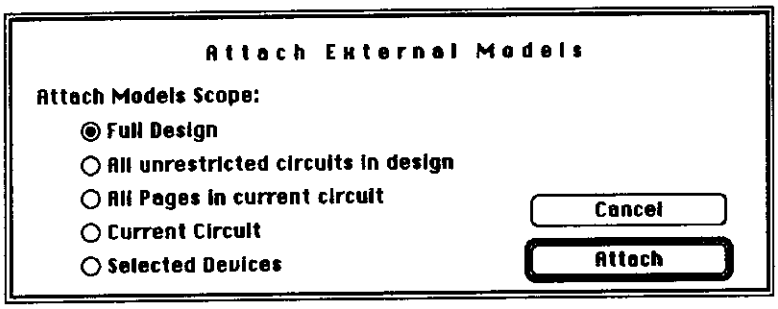
The remove a folder from the search list, simply select it in the list and click on the Remove Folder button.

Selecting Specific Logic Families

The Logic Families selections allow you to restrict the types of devices that will have models attached to them. When the "All Types" button is selected, every device that is used that is a sub-circuit primitive type and does not have an internal sub-circuit will result in a search for a model to attach. When the Select Types button is selected, only the specific standard families indicated will be checked for external models.

Batch Loading External Circuit Models

SimLoad allows the batch loading external models for any group of selected devices or the entire design. The Attach Models button in the SimLoad box will display the following options box:



The Attach Models Scope selections allow you to choose the scope of the search. When the Attach button is clicked, the devices in the given scope are checked, external references are created or updated, where possible, and the external circuits are loaded and attached. After the operation is complete, a summary box will be displayed indicating the number of models found and loaded.

NOTE: This operation can result in a dramatic increase in the amount of memory used by the design. You will be notified if insufficient memory is available to continue.

Purging External Circuit Models

SimLoad's Detach Models operation performs the reverse operation to Attach Models. Devices in the selected scope are scanned and the internal circuit is removed from every device having an external reference. The external reference attributes are left intact, allowing for later reloading of the external models.

Batch Updating or Reloading Models

When an external sub-circuit is loaded, the Schematic tool stores the "Last Modified Date" of the file in the ExtCctDate attribute field of the device. This allows it to check the status of external files relative to a device's internal circuit. Devices that have existing sub-circuits that were loaded from external models, or that have an external reference but no internal circuit, can be updated using the Update External Sub-Circuits command in the Sub-Circuit sub-menu of the Options menu in the Schematic tool. This command will report if all selected items are already up to date.

Model File Naming Conventions

When searching for an external sub-circuit model to attach to a device, the SimLoad tool generates a file name based on the type name, i.e. the name as it appears in the parts palette. The specified search folders are then scanned for the generated file name.

File names are generated differently for each common logic family type, using the following rules:

- File names always end with ".EXT" (note that file names are not case-sensitive).
- Any text in parentheses is ignored, since these are assumed to be package codes.
- If the first two characters of the name are "54" or "74" the a 54XX or 74XX logic family is assumed. Any alphabetic characters are skipped (i.e. the family type) and the file name "74_nnn.EXT" is generated, where "nnn" refers to any remaining digits that are found.
- If the first character of the name is "4" a 4000-series part is assumed. Any alphabetic characters are skipped and the file name "4_nnn.EXT" is generated, where "nnn" refers to any remaining digits that are found.
- If the first character of the name is "1", then a 10K or 100K ECL part is assumed and the name "10_nnn.EXT" or "100_nnn.EXT" is generated, as appropriate.

- For all the above cases, if the name ends with "M", it is assumed to be a "monolith" symbol (i.e. with multiple gates in one symbol) and the M is inserted before the ".EXT", e.g. "74_00M.EXT".
- For all the above cases, if the name ends with a gate unit extension (e.g. "74ALS240.a"), then the unit letter is inserted before the ".EXT", e.g. "74_240a.EXT".
- For all other cases, the type name (with text in parentheses stripped) is used verbatim.

Checking Individual External References

The Attach External Sub-Circuit command in the Sub-Circuit sub-menu of the Options menu in the Schematic tool is useful for checking or editing the external reference attributes of a single device. Select the device in question in the schematic, then select this command. It will show the current status of the external reference and allow you to update it.

Customizing Delays for Logic Families

For cases such as the 74XX families, only one model is provided for all logic families (e.g. 74ALS, 74LS, 74HC, etc.).

> NOTE: These models provide logical functionality only. No attempt is made to match the delay characteristics of the individual parts.

Delays can be manually customized to match a specific part type by either:

- Modifying the sub-circuit model (either in the model file or after it has been used in the design), OR
- Setting appropriate pin delays in the parent device.

Creating External Sub-Circuit Models

External sub-circuit models are simply normal DesignWorks design files with port connectors added to match the pins on the parent device. Following is a brief summary of the considerations affecting sub-circuits:

- In order to be loaded automatically using the SimLoad tool, the file must be named according to the conventions outlined earlier in this chapter.
- A port connector should be placed in the sub-circuit for each pin on the parent symbol, including no-connect pins. Failure to do this will result in the Schematic t[illegible] reporting an error each time the sub-circuit is opened and closed.

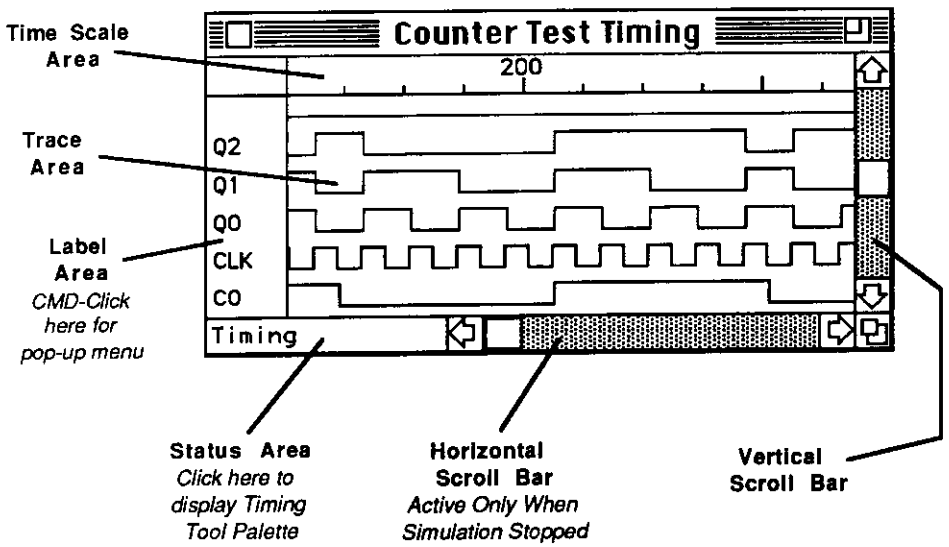
- Each port connector must be named to match the corresponding pin name on the parent device.
- The type of each port connector should match the type of the corresponding pin on the parent device. E.g. An input pin on the parent device should match with a "Port Conn In", or other appropriate type. For more information on pin types, see the section "Working with Hierarchical Blocks" earlier in this chapter.
- The sub-circuit can contain any combination of other devices, including nested sub-circuits to any depth. The only restriction is that the sub-circuit may not contain the parent type or any other type that would create a recursive nesting of blocks.

Other considerations relating to the creation of sub-circuits are discussed in Chapter VII - Hierarchical Design in the DesignWorks/Schematic Reference Manual.

Chapter V - The Timing Tool

The Timing tool allows you to display and edit timing waveforms in graphical form. The timing window is displayed by selecting the Timing command in the Tools menu. Only one timing window can be displayed per design, regardless of the number hierarchy levels in the design.

The Parts of the Timing Window



Time Scale	Located just below the timing window's title bar, the time scale is used to establish the absolute timing of value changes in the trace area. The scale is dependent upon the timing resolution (set using the <> and >< buttons in the timing tool palette). The time scale is also used to set insertion points and selection intervals for use in editing functions.
Trace Area	This area displays simulation results and allows editing of waveforms. Waveforms can only be modified in the future, i.e. times greater than the current simulation time.
Label Area	Displays the list of names of the corresponding to the timing traces at right. Traces can be repositioned by dragging them vertically in this area. In addition, a pop-up trace menu can be displayed by holding the Command (⌘) key pressed while clicking in this area.

Status Area	This box shows progress information for timing commands. Clicking in this box displays the timing tool palette.
Horizontal Scroll Bar	This allows you to display time to the right or left of the present viewing area. The horizontal scroll bar is disabled when the simulation is running.
Vertical Scroll Bar	This will display the signal labels and their corresponding traces above or below the ones presently displayed

Displaying Signals in the Timing Diagram

Adding a Signal Trace

To add one or more signal traces to the timing diagram:

- Select any number of named signals in the schematic.
- Select the Add to Timing command in the Simulation menu.

Removing a Signal Trace

To remove a trace from the timing diagram:

- Hold the Command (⌘) key and click on the name in the label area at the left side of the timing window.
- Select the Remove command in the pop-up menu.

You can remove multiple traces in one operation by holding the Shift key to select multiple labels, then selecting the Remove command;

Repositioning Traces

Any collection of selected labels and their corresponding timing traces can be repositioned within the list by clicking on the desired names (using the Shift key if desired to select more items) and dragging the outlined box vertically to its new location. Releasing the mouse button will cause the list to be revised with the labels and traces in their new positions. Alternaltively, the To Top, To Bottom and Collect commands in the timing pop-up menu can be used.

Timing Display Groups

The Timing tool allows multiple signal lines to be grouped into a single trace with values displayed in hexadecimal.

Creating a Group Trace

A group trace can be created by any of the following methods:

- Select any collection of traces by SHIFT-clicking in the label area, then select the Group command in the timing pop-up menu (i.e. by holding the command (⌘) key while clicking in the label area).
- Select any collection of signals in the schematic diagram, then select the Add As Group command in the Simulation menu.

NOTE: If any of the selected traces is already displayed, it will not be added to the group.

- Select a bus in the schematic diagram, then select either the Add To Timing or Add As Group command in the Simulation menu. Busses are added as a group by default. They can then be ungrouped if desired, using the Ungroup command in the timing pop-up menu.

Order Within a Group

For the purposes of displaying a hexadecimal value for a group, the order of signals within the group is important. When a group is created, the following rules are used to establish the order:

- If the signal name has a numeric part (e.g. "D12" or "WRDAT4X", then the numeric part is used to sort the signals. The lowest numbered signal will be the least significant bit of the group value. Any unnumbered signals will be in the most significant bit positions.
- Otherwise, the signal's existing position is used, i.e. traces that appeared higher in the timing diagram will be more significant.

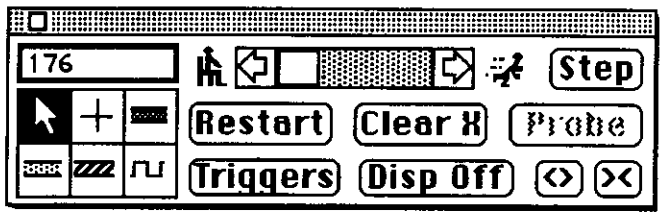
The order of signals within a group can be changed using the Get Info command on a group trace. This is displayed by selecting the Get Info command in the timing pop-up menu or by double-clicking on the label.

Entering a Group Name

When a group is first created, a group name is automatically generated from the names of the enclosed signals. This name can be edited using the Get Info command in the timing pop-up menu.

NOTE: The group name is lost when an Ungroup operation is done.

The Timing Tool Palette



Displaying and Hiding the Tool Palette

The timing tool palette is displayed by default when a new timing window is created. It can be hidden by selecting the Hide Timing Palette command in the Simulation menu or by clicking in its "go-away" box at the top left corner. To redisplay the palette, select the Show Timing Palette command in the Simulation menu or click in the timing status area at the lower left corner of the timing window.

NOTE: The timing tool palette can be displayed even if there is no timing window displayed. This allows you to make use of the simulation controls even if you are not using the timing diagram.

Tool Palette Time Display

The status box at the top left of the tool palette displays one of two different time values, depending upon the status of the simulator:

- If the simulator is stopped and the cursor is positioned in the time scale or trace area of the timing window, it shows the time corresponding to the cursor position.
- Otherwise, it shows the current simulation time as the simulation progresses. It cannot be disabled.

Tool Palette Controls

The control buttons in the tool palette provide quick access to functions alternatively located in the Simulation menu.

...	This speed control allows graphical selection of a simulator speed. Clicking in the gray area of the control switches between Run and Stop. Clicking in the arrows or dragging the position box will select an intermediate speed.

Step	Causes the simulator to execute one time step.
Probe	This item is active only when the schematic is frontmost. This activates the schematic signal probe mode.
Restart	Same as Clear Simulation menu command. Clears all pending events, sets time to zero and recalculates all device states.
Clear X	Same as Clear Unknowns menu command. Clears all storage devices and attempts to clear feedback paths in circuit.
Triggers	Same as Triggers menu command. Displays the trigger control box. See the section on Triggers in Chapter III - Simulation.
Disp Off/Disp On	Enables and disables the timing display. Simulation proceeds substantially faster when the timing display is disabled. It can be re-enabled either manually or by a trigger when desired time or signal conditions are met.
<> (Zoom In)	Increases horizontal display resolution, i.e. decrease number of time units per screen dot.
>< (Zoom Out)	Decreases horizontal display resolution so more elapsed time can be viewed in the display.
Drawing Tools	The drawing tools are used for drawing and editing timing waveforms. These are described in the following section.

Timing Diagram Editing

The drawing tools in the timing tool palette and the commands in the Edit menu can be used to graphically create and edit timing waveforms.

NOTE: Timing traces can only be editing in the future, i.e. at times greater than the current simulation time.

Selecting Data for Copy/Paste Operations

To select timing data for the Cut, Copy, Paste, Clear, Duplicate, Insert Time or Delete Time operations:

- Simulation must be Stopped. To do this, use the speed control in the Timing Tools Palette, select the Stop command in the Simulation Speed sub-menu, or click anywhere in the timing window.

• Cursor must be in 'Point' mode. If not already in point mode click on the Arrow symbol in the Timing Tool Palette, or select the Point command in the Edit menu. The cursor will now be an arrow.

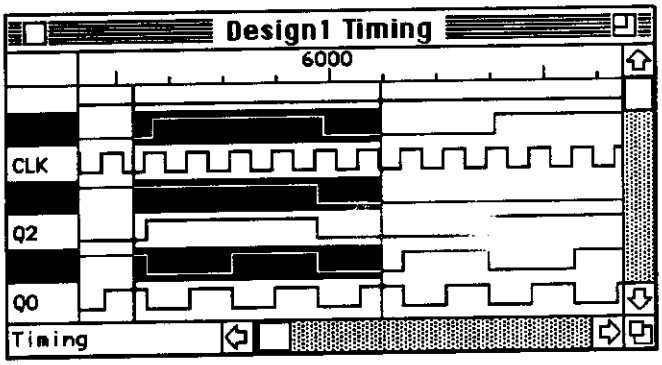
There are two methods of selecting areas for edit operations.

Separate Label and Interval Selection

Using this method, the traces to be affected are selected by SHIFT-clicking in the label area, then the time interval is selected by clicking and dragging in the time scale. This allows you to select non-contiguous traces in the display.

• Click on the desired label in the label area to highlight and select it. To select more than one label, hold the shift key and click on the labels.

• To set the selection interval, click and hold down the mouse button in the time scale at either end of the desired interval. Drag left or right until the desired interval is enclosed. When the mouse button is released the select interval is set and two selection interval lines will appear. If any of the signal labels were selected the timing signal with in the selected interval will be highlighted in time display.



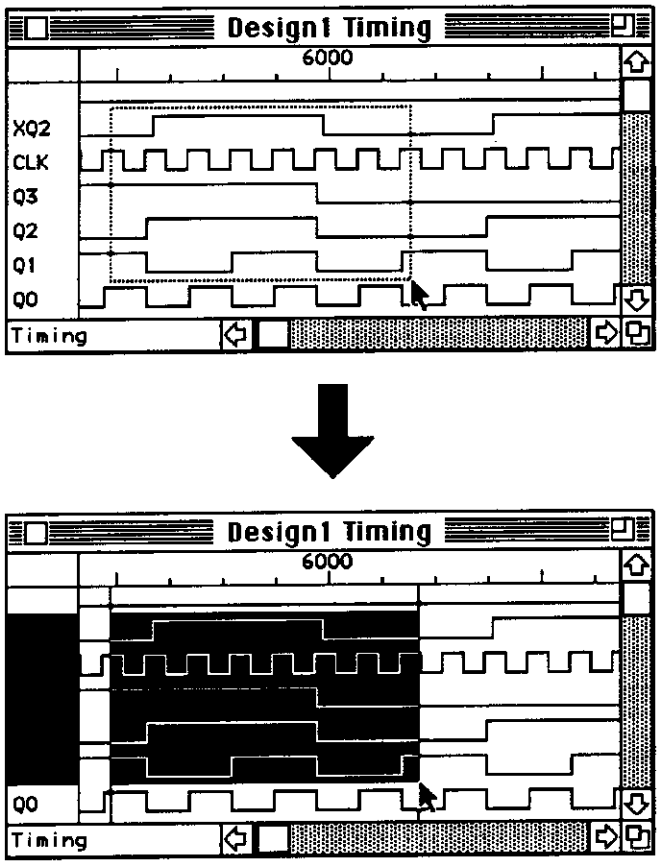
NOTE: Clicking and releasing the mouse button at one spot will create a zero-width interval. This can be used to insert Pasted data without deleting any existing data.

Drag Selection

This method allows you to select a group of labels and a time interval in a rectangular area of the timing diagram.

To do a drag selection, click and hold the mouse button at any corner of the rectangular area you wish to select. Drag diagonally across the desired area.

When the mouse button is released, the enclosed time interval and traces will be selected.



NOTE: The selection operations in the timing window have no effect on selections in the schematic window.

Selecting All Traces or All Time

To select a specific time interval in all traces on the diagram:

- Use the Select All command in the Edit menu to select the entire diagram.
- Drag select an interval in the time scale area without clicking in the trace area.

To select all time for specific traces:

- Use the Select All command in the Edit menu to select the entire diagram.
- Click at the top of the label area, above the highest label displayed (this will deselect all traces), then shift-click to select the desired traces.

Deselecting

Clicking anywhere in the trace area not in a trace will deselect the labels and selection interval.

Clicking in the label area above or below the label list deselects all traces but leaves the current interval selected.

Summary of Timing Edit Commands

The following table summarizes operation of the timing edit commands in the Edit menu.

See Chapter IX - Menu Reference for a detailed description of these commands and Appendix A - Timing Text Data Format for a description of the clipboard data format.

Cut	This command copies to the clipboard any signal change events on selected signals in the selected time interval and clears the selected interval. The data is stored in both picture form and text data form. Events after the selected interval are not moved forward. The Delete Time command can be used to do this.
Copy	The Copy command copies the selected timing data to the clipboard in picture and text format. See the notes under the Cut command, above. Note that Copy can be used on a selection to the left of (older than) the current simulation time since it does not modify the selected data.
Paste	The Paste command pastes the text timing data from the clipboard onto the selected area of the timing diagram. The selected time interval is deleted and then the new data is inserted. I.E. data following the selection interval will be moved forward by the width of the selection interval, then back by the width of the pasted data.
Clear	The Clear command clears any signal change events in the selected area on the timing diagram. Later events are not moved forward.

Duplicate	This command inserts a duplicate of all selected signal data on the timing diagram after the selected interval. I.e. the selected data is copied to a temporary location, then the selection point is moved to the end of the selected interval and the copied data is inserted at this point. All signal changes after the duplicate data are moved back in time by the width of the original selection.
Insert Time	This command inserts a blank time interval in the selected traces. The new interval is inserted in front of the selected interval and is of the same width as the selected interval.
Delete Time	This command deletes the selected time interval from the selected traces and moves all later data ahead by the width of the interval.

NOTES:

1) If you wish to paste a timing picture into a word processing package, it may be necessary to paste it first into a drawing program to select the picture data on the clipboard. A word processing package will normally take the text data from the clipboard by default.

2) You cannot modify timing data that is older than the current simulation time.

The Drawing Tools

The timing tools are used in editing the signal lines displayed in the timing window. New states can only be created and modified in the future (to the right of the point in time where the simulation was halted).

NOTES:

1) The simulator must be stopped to perform any timing editing operations.

2) The drawing tools have no effect on grouped lines. A group can be ungrouped into individual signals for editing, if desired, using the Ungroup command in the timing pop-up menu.

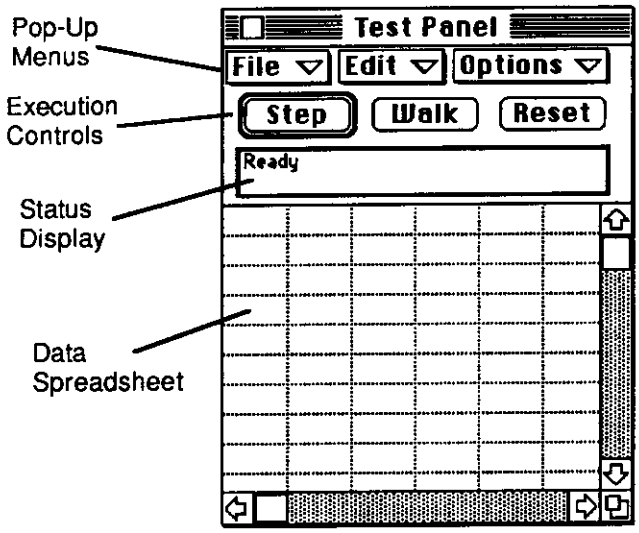
	High/Low	• Creates the a high or low signal state, depending the vertical position the cursor after dragging and releasing the mouse button. • Modifies the time at which a change occurs by clicking on and dragging horizontally the vertical edge of the signal and is set at the time where the mouse button is released. • Modifies the state by clicking on a horizontal edge and dragging up or down. The new state is set when the mouse button is released.
	Don't Know	• Creates the don't know signal state. • Modifies the time at which the state occurs by clicking on and dragging horizontally the vertical edge of the signal and is set at the time where the mouse button is released.
	High Z	• Creates the high impedance signal state. • Modifies the time at which the state occurs by clicking on and dragging horizontally the vertical edge of the signal and is set at the time where the mouse button is released.
	Conflict	• Creates the conflict signal state. • Modifies the time at which the state occurs by clicking on and dragging horizontally the vertical edge of the signal and is set at the time where the mouse button is released.
	Insertion	• Inserts only high/low signal states in between previously created states.

Chapter VI - The TestPanel Tool

The TestPanel tool allows you to interactively step through test vectors (i.e. lists of signal values) and monitor simulation results in a spreadsheet-style format.

General Principles

When the TestPanel is invoked (by selecting its name in the Tools menu), you will see a number controls across the top plus a spreadsheet-like data section across the bottom.



Pop-Up Menus

Each TestPanel window has its own File, Edit and Options menu. TestPanel does not respond to the menus in the main menu bar at the top of the screen.

See Chapter IX - Menu Reference for a detailed description of the TestPanel menus.

TestPanel Execution Controls

The Step, Walk and Reset buttons control basic operation of TestPanel and are described later in this chapter.

Status Display

The status display area shows the current operational status of this TestPanel window.

The Data Spreadsheet

The data spreadsheet section is used to enter your "test program", i.e. a sequence of signal values to apply to the circuit under test, and to view the resulting signal changes.

Time normally progresses downward, i.e. a single row of the spreadsheet represents one simulator time step. Once all these values are set and commands executed, the next line is selected.

NOTE: The Options command in the Options pop-up menu allows you to select an "across" data orientation. This swaps the two axes of the spreadsheet but otherwise operates identically. To simplify wording, all descriptions of TestPanel operation will assume we are in the default "down" orientation.

Data Entry

Data is entered by clicking in the desired cell and typing directly onto the spreadsheet. Cells can contain signal value data and commands which indicate which signals to deal with and control execution of the test program. Commands always start with a "$" character to distinguish them from data.

In the simplest case, you can enter a sequence of signal values in one column and view the resulting outputs in another column. You can step through the test data interactively using the "Step" or "Walk" buttons, or set up all the events specified in your test program in one block, using the Copy To Timing menu command.

Each of the signal value columns should begin with a "header", i.e. a command listing the signals to be used in that column and whether they are inputs or outputs. If no header is specified in a given column, that column will be ignored and its contents are effectively comments.

Modes of Operation

The TestPanel tool has three basic modes of operation:

- In "interactive mode" (the default), the test vectors and instructions entered in the spreadsheet are evaluated, new signal values are set up in the simulated circuit, and resulting outputs are displayed before proceeding. This mode gives a great deal of interaction and control over the simulation, but at the expense of execution speed. Interactive mode is controlled using the "Step" and "Walk" buttons.
- In "monitor mode", the TestPanel acts as a passive display window for simulation results. In this mode, the circuit is checked after each simulated time step and any desired signal values are displayed in the spreadsheet. Monitor mode is invoked by selecting the Monitor Mode menu command.
- In "copy to timing" mode the data in the data spreadsheet are evaluated and a sequence of signal value changes are set up in the simulator corresponding to all the test vectors. The TestPanel then allows the simulator to proceed to evaluate the scheduled inputs without further interaction. This method of setting up simulation inputs allows the simulator to proceed at full speed, but gives no user interaction during the simulation (of course, the normal DesignWorks interaction with the circuit and the timing diagram are still possible). This process is invoked by selecting the Copy to Timing command.

TestPanel and Hierarchical Designs

TestPanel can be used with hierarchical designs, but each TestPanel window can be associated with only one circuit or sub-circuit at any given time. Multiple TestPanel windows can be open at one time and associated with different sub-circuits. Hierarchical signal names (i.e. prefixed with the names of parent devices) are not supported.

TestPanel-to-Circuit Association

When a TestPanel window is opened, it by default links to the current circuit (i.e. the one displayed in the frontmost document window). It will then search that one circuit for any signal names specified in the test program and perform operation only on that circuit.

It will remain linked to that circuit as long as the circuit is open, even if it is no longer the frontmost. If the circuit is closed, the TestPanel will become inactive until it is attached to a new circuit using the Link to Circuit command in the Options pop-up menu.

NOTE: TestPanel can only be used on circuits that are open for editing.

Multiple TestPanels

Multiple TestPanel windows can be operated concurrently with completely independent operating modes, sets of test vectors and circuit associations. For example, one panel could be used to generate a repeating test pattern while another monitors outputs on a set of signals in a different sub-circuit block.

This is done by simply selecting the TestPanel item in the Tools menu while another TestPanel window is already open. However, we recommend using this technique only once you are quite familiar with general TestPanel operation.

Interaction With Other Simulation Tools

The TestPanel can be used in conjunction with the Timing tool and other simulation display and control mechanisms, however the following points should be noted:

- Results will be unpredictable if two tools are attempting to modify the value of the same signal. This can be caused by the TestPanel modifying a signal that has also been drawn on the timing diagram, multiple TestPanels driving the same signal, or attempting to change the value of a signal that is being driven by a device on the schematic.
- By default, the TestPanel proceeds to the next step in the test program only after all signal states in the design have settled (i.e. there are no signal change events pending). If the design contains any clock devices or oscillations, or if some other tool is creating signal change events, the design may never settle and the next step will never be executed.

Using TestPanel

Opening a TestPanel Window

The TestPanel tool is invoked by selecting the name TestPanel in the Tools menu. The TestPanel window can be left open while working on your schematic, but if any schematic editing is done, the tool will perform an automatic Reset. It is recommended that you close the TestPanel while doing general schematic editing.

Closing TestPanel

You can close the TestPanel window by clicking in the "go-away" box at the top left corner of the window. If any changes have been made in the data spreadsheet and not saved using the File menu commands you will be prompted to save the data.

Editing in the Spreadsheet

IMPORTANT NOTE: The TestPanel has its own pop-up menus for editing commands and does not respond to the commands in the main menu bar at the top of the screen. These will continue to affect the current document window.

Editing Text in a Cell

Text in a cell can be edited at any time simply by clicking in the cell and typing. Standard Macintosh text editing rules apply. Only a single line of text can be typed in a cell.

Undo

Undo is not supported in TestPanel.

Selecting Multiple Cells

Any rectangular block of cells can be selected by clicking in the cell at one corner of the block, then holding the Shift key and clicking at the opposite corner. The selected cells can then be operated on using the commands in the TestPanel's Edit pop-up menu. Drag selection is not supported.

Moving Between Cells

Keyboard operations can be used to move between cells while entering data, as summarized in the following table.

Tab	Moves to the next cell to the right.
Return	Moves to the cell under the last one clicked in.
Arrows	One cell in the given direction.

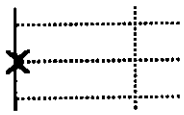
Clipboard Operations

The standard Edit menu operations Cut, Copy, Paste and Clear can be used on any block of selected cells or on a segment of text from a single cell.

See the section TestPanel Menus later in this chapter for a detailed description of these commands.

Resizing rows and columns

Rows and columns of the spreadsheet section can be resized by clicking and dragging on the dividing lines between them anywhere close to the edge of the viewing area. You will see the cursor change shape to a ✕, as in the following:



TestPanel Controls

Step

The Step button causes TestPanel to interactively execute the next line of the data spreadsheet, i.e. the line which contains the currently selected cell. If the line selected is past the last data line, then a beep will be issued and no action will be taken.

Walk

The Walk button causes the TestPanel to step through the data spreadsheet as quickly as possible in interactive mode, starting with the currently selected cell. To start from the beginning of the program, it is necessary to select a cell in the first row of the spreadsheet, or first click the Reset button.

Reset

The Reset button removes all data that was created by the program from the spreadsheet and selects the top-left cell. All user-entered data (i.e. headers, commands, expected values, etc.) are left intact.

The Test Program

The spreadsheet area forming the lower part of the TestPanel window is where all test vector data is entered and results are displayed.

NOTE: The Options menu command allows you to specify an "across" (i.e. time axis is horizontal) or "down" orientation of data in the spreadsheet. The default "down" orientation will be assumed for all discussions that follow. If you are using the "across" orientation, the you will have to swap the terms "row" and "column", etc., in these discussions.

Row and Column Usage

In the spreadsheet area, each horizontal row usually corresponds to one "test vector", i.e. a set of signal stimulus and comparison data that is applied at a particular time step in a simulation. Until the limit of 32767 rows is reached, you can always enter new data into the empty rows displayed at the bottom of the spreadsheet.

The cells in the spreadsheet have no fixed meaning. Their usage at any given time depends upon the last command that was encountered in that column. Normally, we recommend using separate columns for inputs, outputs, and commands, but it is quite possible to mix them in a single column.

$Commands

TestPanel commands always consist of a "$" character followed by a keyword. Commands fall into two groups:

- *Definition Commands* indicate how the following cells in the same column are to be interpreted. These usually give a list of signal names whose values will be specified or observed. Definition commands have no immediate effect on the execution of the test program or on the circuit under test.
- *Control Commands* allow specific loop control, delay or other control actions to be specified.

NOTE: For compactness, command keywords can always be shortened to the minimum number of letters needed to distinguish them from other commands. In most cases, only one letter is needed, e.g. "$I" can be used instead of "$INPUTS". The minimum contraction is shown in the command list below.

Note the following rules concerning command execution:

- Commands have to be "executed" (i.e. stepped through) before they take effect. E.g. if you select the cell under a command and then step downwards from there, the command will have no effect.
- Non-command cells (i.e. anything not starting with a "$") will be ignored completely unless a previous definition command indicates how to interpret them.
- A definition command in a given column always completely overrides any previous definition command in the same column.
- If the circuit schematic is modified while a test program is in progress, all current signal lists are invalidated and no signal data will be interpreted or displayed until the next definition command is encountered.

Definition Commands

NOTE: The $DELAY command can be used either as a control command (if a delay value follows the command keyword) or definition command (if the $DELAY command appears alone in a cell). It appears in both sections below.

Command	Short Form	Description
$TIME	$T	Display current simulation time in the following cells.
$INPUTS signalList	$I	Specify the values of input signals.
$OUTPUTS signalList	$O	View the values of the given signals (can actually be used to view the "real" values of inputs as well).
$EXPECTED signalList	$E	Specify the expected values of the given signals
$WAIT signalList	$W	Pause the test program until a specified value is detected.
$DELAY	$D	Delay by the specified amount before executing this line.

Signal List Format

Many of the definition commands require that a list of signals be specified. The format of this list can be seen in the following examples:

```
CLK
```

specifies simply the single signal "CLK". In this case, each test vector will consist of a single entry, e.g. "0" or "1".

```
A B C D
```

specifies four separate signals A, B, C and D. In this case, each test vector will consist of four values separated by spaces, e.g. "0 1 1 0".

```
[A B C D]
```

uses the grouping operator [] to indicate that the four signals will be specified by a single hexadecimal number with A in the most significant bit position. E.g. the hex test vector "E" (or 1110 binary) would assign a 1 value to A, B, and C, and a 0 value to D.

NOTE: For compatibility with the Timing tool, TestPanel will allow a "group name" to be prefixed to a group, e.g. "CONTROL[A B C D]". This group name is not used internally and will be ignored. For this reason, there must be a blank between any preceding signal name and the opening "[" or the name will be ignored.

```
[D3..0]
```

uses the ".." operator to indicate a sequence of numbered signals. This case also uses the [] to indicate that the four signals D0, D1, D2 and D3 are to be treated as a single integer, in this case with D3 in the most significant position.

```
CLK CLR [D15..0] [FC2..0]
```

specifies four groups, the first being the signal CLK, the second being the signal CLR, the third being the numbered signals D0 through D15, and the last being the set FC0, FC1 and FC2. A typical test vector for this group might be:

```
.C 0 ++ 5
```

In this case, the ".C" specifies that a clock pulse is to be applied to the CLK signal, the value 0 (low) to the CLR, the next sequential integer value to the 16 signals D0, D1, etc. and the hex value "5" (or 101 binary) to the signals FC2, FC1 and FC0. See definitions of these codes below.

The $INPUTS Command

All cells below this command in the same column are used to specify the input values to be set up in the circuit under test. As each line of the test program is read, the cell in this column is interpreted and new values are placed on the signal lines in the circuit. This command must be placed in

the first row of this column to specify which signals are to be set up. The values specified in all subsequent rows are interpreted according to that header. See the information on header format above.

All non-command cells in this column are interpreted as signal value data. A data cell should contain one signal value item for each signal group specified in the most recent header (a signal group being either a single signal, or a group specified in brackets [] in the header. If the cell is empty or if less items are specified in the data cell than in the header, then unspecified items will be left unchanged from the previous time they were specified.

Each signal value item can be one of the items in the following table.

A hexadecimal value	This will be interpreted as an integer with one bit for each signal in the corresponding header group. For individual signals, only the least significant bit is used, so the only values will be 0 or 1. For multiple-signal groups (i.e. specified in []'s in the header), the last signal in the group will take the value of the least significant bit and preceding signals will take the values of successive higher numbered bits. Unused high-order bits are ignored.
The value code ".C" or ".c"	This code indicates a positive clock pulse (i.e. a transition to the 1 level, followed by a transition to the 0 level), with a width as specified in the Test Options command. Note that if the signal is already at a 1 level, no rising transition will occur. See also the notes below.
The value code ".N" or ".n"	This code indicates a negative clock pulse (i.e. a transition to the 0 level, followed by a transition to the 1 level), with a width as specified in the Test Options command. Note that if the signal is already at a 0 level, no falling transition will occur. See also the notes below.
The value code ".Z" or ".z"	This code indicates a high impedance value.
The value code ".X" or ".x"	This code indicates a "Don't Know" value and can be used to test a circuit's response to unknown inputs.
The increment operator "++" or "++n"	This code indicates that the integer value specified previously is to be incremented and the new value applied. If an integer is supplied immediately following the "++" then this number is added to the old input value, otherwise 1 is used.

The decrement operator "--" or "--n"	This code indicates that the integer value specified previously is to be decremented and the new value applied. If an integer is supplied immediately following the "--" then this number is subtracted from the old input value, otherwise 1 is used.

NOTE: Care must be taken regarding setup times when using the .C or .N clock pulse operators. E.g. .C generates a rising transition immediately which may not allow any setup time after values settled from the last step. This situation can be avoided in most cases by using .N to test positive-edge-triggered devices and .C for negative-edge-triggered devices. This ensures that a setup time at least the width of the clock pulse is allowed.

Command examples:

```
$I [D3..0]
$I CLK CLR LD/
$I FC0..2
```

The $EXPECTED Command

This command indicates that the following cells will contain the expected values for signals in the circuit under test. The values in the circuit are compared to those specified. The result of this comparison is shown as a mark character in each data cell this command. If the actual values equal the expected ones, then an "=" is displayed in the cell, otherwise a "≠" is displayed. If no value appears in a data cell, then no comparison is done.

Each signal value item can be one of the items in the following table.

A hexadecimal value	This will be interpreted as an integer with one bit for each signal in the corresponding header group. For individual signals, only the least significant bit is used, so the only values will be 0 or 1. For multiple-signal groups (i.e. specified in []'s in the header), the last signal in the group will take the value of the least significant bit and preceding signals will take the values of successive higher numbered bits. Unused high-order bits are ignored.
The value code ".X" or ".x"	This code indicates a "Don't Know" value and can be used to test a circuit's response to unknown inputs.
The value code ".Z" or ".z"	This code indicates a high impedance value.

The value code ".DC" or ".dc"	This code indicates a "Don't Care", i.e. it will always match. Note that this is different than not specifying a value, in that this will display an "=" result, whereas no comparison is done if no value is given.

The comparison is done either at the same time step that the inputs are set, or after waiting for the simulation to settle (see the Immediately/After Values Settle in the Options command described elsewhere).

Command examples:

```
$E [ADDR0..15]
$E EQ
$E A B C D
```

The $OUTPUTS Command

This command specifies that the following cells should show the actual values read from the circuit and the result of the comparison with the expected values. Any cell in this column not containing a command will be overwritten when the outputs are read.

Command examples:

```
$O [ADDR0..15]
$O EQ
$O A B C D
```

The $TIME Command

This command specifies that the following cells should show the simulation time when execution of this step is started. Time values are displayed in decimal.

Command examples:

```
$TIME
```

The $WAIT Command

This command specifies that the following cells will contain signal value conditions to wait for before proceeding. This command is similar to $EXPECTED, except that instead of display an equal or non-equal status, the test program waits until an equal status occurs. Note the following rules:

- If the wait condition is already met, then the test program proceeds and the wait has no effect.

- The test program enters the wait state after any new input values specified in the row have been set up.
- If any delay is specified in the same row, then the delay applies concurrently and it becomes, in effect, a minimum wait time.

Command examples:

```
$WAIT SYNC
$WAIT [D7..0]
```

The $DELAY Command

This command specifies that the following cells in the same column should be scanned for a decimal integer delay value. The following rules are applied:

- If a value is found, the specified delay is applied after setting up any input value specified in this row. I.e. This specifies the duration of any values applied on this line.
- If multiple $DELAYS appear in one row, then only the longest will apply.
- If no value appears in this column, then the default delay (specified in the Test Options command) will apply.

Command examples:

```
$DELAY
```

NOTE: If any value appears after the $DELAY keyword, then it is taken to be a control command and the following cells in the column will not be interpreted as delays. See the description of this format below.

Control Commands

Command	Short Form	Description
$REPEAT or $REPEAT nnn	$R	Repeat rows up until the next $END either indefinitely or nnn times.
$END	$EN	Marks the end of a REPEAT loop.
$DELAY nnn	$D	Delay for nnn time units after applying the input values specified in this row.
$STOP	$S	Stop execution after this row.

$REPEAT Command

The $REPEAT/$END construct allows you to specify any integer number, or infinite, repetition of a group of test vectors. The format of the $REPEAT command can be seen in the following example:

	$IN CLK	$IN CLR	$IN [D3..0]	$OUT [Q3..0]
	0	0	0	
	0	1	0	
	0	0	0	
$REP 32	0	0	++	
$END	1	0		

In this case, all vectors from the line containing the REPEAT down to and including the line containing the END will be repeated 32 times. If the repetition factor is omitted, it will be taken as infinite in interactive mode, and 1 in timing setup mode.

REPEATs can be nested to any desired depth. If the END is omitted, the last line in the data spreadsheet that has been modified is taken as the end of the loop.

$DELAY Command

When the $DELAY command appears with a value after it, it is taken as a control command and the specified delay is applied after execution of the current row. See the description of the $DELAY definition command above for more details.

Command example:

```
$D 15
```

$STOP Command

This command causes the test program to cease execution after applying any new input values specified in this row.

TestPanel Menus

File Menu Commands

Open Test File

This menu item allows you to load the data spreadsheet from a text file which was saved during a previous session or created externally.

The format of this file is the same as that commonly used for transferring data between spreadsheet applications:

- the text from the top cell in the leftmost column is written first, followed by a tab character, followed by the contents of the next cell to the right, followed by a tab, etc.
- each line is terminated by a carriage return character.

Save Test File

This menu command saves the contents of the data spreadsheet to a text file in the format described above.

Set Trace File

The Set Trace File button allows you to create a text file to which will be written a line of text data matching the current line of the spreadsheet, each time a step is done in interactive or monitor mode. This allows you to create a permanent record of simulation data. This file can be opened and closed at any time during the simulation without disrupting any simulation data.

Save to Design Attr

This command behaves identically to Save Test File As except that the destination for the data is a design attribute field instead of a file. The TestPanel data will then be stored with the design file the next time the design itself is saved.

This command will display a list of the fields defined for the current design allowing you to select one as the destination. All existing data in the

selected field will be lost. Note that attribute fields are limited to 32,000 characters.

Load From Design Attr

This command corresponds to the Open Test File command except that the data is loaded from a design attribute field instead of a file.

Edit Menu Commands

Undo

Undo is not supported in the TestPanel.

Selecting Multiple Cells for Clipboard Operations

The clipboard operations Cut, Copy and Paste can operate on the selected characters in a single cell or on any rectangular group of cells in one operation. To select a group of cells, click at the top-left corner of the group, then hold the Shift key depressed while clicking at the bottom-right corner. All the cells in the group will be highlighted.

NOTE: The Paste command does not require a group to be selected. It always pastes all the data on the clipboard starting at the top-left cell currently selected. The size of the selected area is not significant.

Exchanging Clipboard Data With The Timing Tool

The Timing tool uses a clipboard data format which is a subset of the command structure supported by TestPanel. Therefore, data copied from a timing diagram can always be pasted into a TestPanel window. Conversely, a TestPanel test program can be copied and pasted into the Timing window as long as it is restricted in format. In general, the Timing tool ignores all control commands such as $REPEAT and $WAIT, as well as any expected output specifications.

See Appendix A - Timing Text Data Format for a complete description of the Timing clipboard format.

Data Format for Clipboard Operations

The Cut, Copy and Paste commands operate on text data in the following form:

- text for top-left selected cell, followed by a tab character, followed by text for next cell to the right, followed by a tab, etc. to the end of the selected cells in the row.
- the last cell in a row is followed by a carriage return character instead of a tab.
- if a selected cell is empty, the preceding and following tab characters will indicate its place.

Cut

The action of the Cut command depends upon how data is selected in the spreadsheet. If a group of cells has been selected using the Shift key, then the contents of all selected cells will be copied to the clipboard in the format described above, then the cells will be cleared.

If a cell is being actively edited (i.e. a portion of the text is selected or the I-beam cursor is flashing in it) then only the selected text will be copied to the clipboard and the selected text will be deleted.

Copy

The Copy command operates identically to the Cut command except that the selected items are not cleared.

Paste

The Paste command copies the data on the clipboard to the spreadsheet. Data is assumed to be in the clipboard format described above, i.e. any number of cells can be pasted in one operation.

Note the following rules for the Paste operation:

- The size of the selected area on the spreadsheet has no effect on the Paste operation. The top-left cell in the selected group will be the top-left corner of the area affected by the Paste. However, the number of cells affected by the Paste is determined entirely by the data on the clipboard.
- Tab or carriage return characters cannot be pasted into TestPanel cells since they are used to delimit cell data.

- If a cell is being actively edited at the time of the Paste (i.e. a portion of its text is selected or the blinking I-beam cursor is located in it) then the text on the clipboard up to the first tab will replace the selected text in that cell.
- Data is always copied and pasted as viewed on the screen without regard for the "across" or "down" data orientation TestPanel setting.
- Clipboard operations cannot be undone.

Clear

This command clears all text data from the selected cells. THIS CANNOT BE UNDONE!

> Note: To clear only specific types of cells throughout the spreadsheet area, use the Clear Sheet command in the Options pop-up menu.

Select All

This command selects the entire spreadsheet area.

Fill Down

This command copies the data from the topmost cell in each selected column to all selected cells underneath it. It has no effect on the clipboard.

Fill Right

This command copies the data from the leftmost selected cell in each row to all selected cells to the right of it. It has no effect on the clipboard.

Insert Rows

This command inserts above the topmost selected row a number of rows equal to the number currently selected. All lower rows are moved down.

> NOTE: This affects the entire width of the spreadsheet, regardless of the width of the selected area.

Delete Rows

This command deletes the entire width of each row touched by the current selection or containing the cell currently being edited. All data in the selected rows will be lost and all lower rows will be moved up. THIS CANNOT BE UNDONE!

Insert Cols

This command inserts left of the leftmost selected column a number of columns equal to the number currently selected. The columns to the right are shifted right.

> NOTE: This affects the entire height of the spreadsheet, regardless of the height of the selected area.

Delete Cols

This command deletes the entire height of each column touched by the current selection or containing the cell currently being edited. All data in these columns will be lost and columns to the right will be shifted left. THIS CANNOT BE UNDONE!

Options Menu Commands

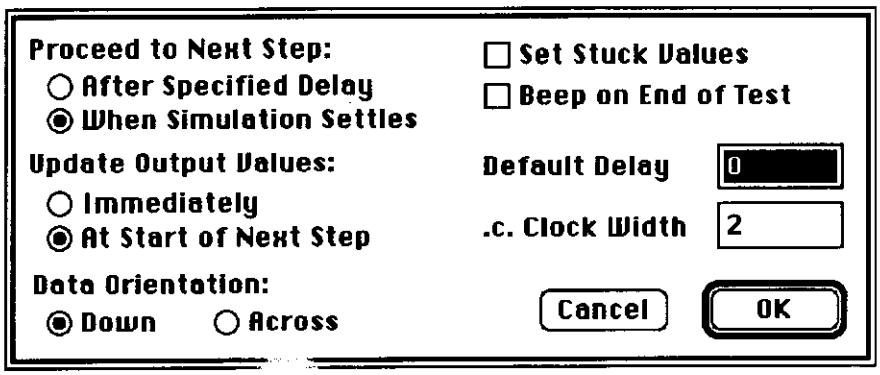
Link to Circuit

This command causes the TestPanel to become associated with the current circuit, i.e. the circuit displayed in the frontmost document window. This is intended primarily for use in hierarchical designs where sub-circuits are being opened and closed.

> *See the notes earlier in this chapter regarding TestPanel and hierarchical designs.*

Test Options

This command will display the following options box:



Proceed to Next Step

These two buttons allow you to determine whether the TestPanel waits for the simulator to complete all processing before proceeding or goes ahead at a fixed pace without regard for other activity.

After Specified Delay	With this setting, the test program will proceed to the next step after the delay specified on that row (if any) or after the Default Delay specified in this box. This setting is most appropriate when using the TestPanel to generate continuous patterns or using it with a design that generates its own clock signals.
After Simulation Settles	With this setting, the TestPanel in effect "goes to sleep" until the DesignWorks simulator has completely settled, i.e. all events have been evaluated and no new ones have been created by the circuit. Note that, depending on the circuit, this may take an arbitrary amount of time. If the simulated circuit has any oscillating elements then this settling will never occur. This setting is preferable for cases where the TestPanel is generating all inputs and clocks, since it allows the test program to be independent of delays in the design.

Update Output Values

These two buttons allow you to determine when the outputs from the design are evaluated.

Immediately	The expected outputs are checked and actual outputs are displayed at the same time step as the inputs were set up, i.e. these will be the values resulting from the previous vector.
At Start of Next Step	Outputs on each line are checked immediately before the next step is executed, i.e. after all simulation values have settled from this step or the specified delay has elapsed. Thus, the output values shown on each row will be the result of the inputs specified in that row.

Data Orientation

These two buttons allow you to select whether time proceeds downwards (i.e. each row represents one time step) or across (each column represents one time step).

If this setting is changed, you will be asked whether you want to swap all data in the spreadsheet area.

Except for the reversing of row and column functions, all TestPanel operations are identical in either mode.

Set Stuck Values

When this checkbox is set, all signal values set up by the TestPanel are "stuck", i.e. they override any outputs from existing devices in the circuit and can only be changed by further TestPanel outputs or other user settings.

Beep at End of Test

When this switch is enabled, a standard beep tone will be issued whenever the end of the test program is reached.

Clock Width

This editable text item determines the width of the clock pulse generated by a ".C" or ".N" signal value in the "Specified Inputs" column. A zero value is not allowed.

Default Delay

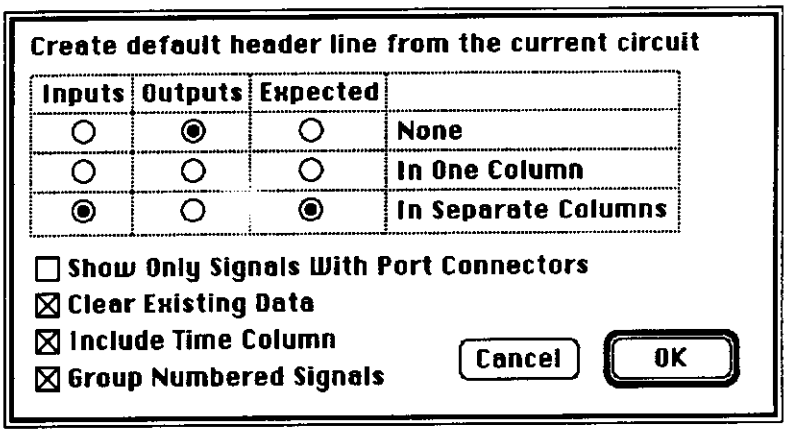
This editable text item operates slightly differently depending on other settings:

- If the "After Specified Delay" step option is selected or a "Copy to Timing" operation is being done, this will be the default delay between steps and can be overridden by a $DELAY command in the test program. A value of zero is not allowed.
- If the "After Simulation Settles" option is selected, this will be the minimum step size. I.e. The time between steps will be the larger of this value or the amount of time required for the circuit to settle. A value of zero is allowed in this mode.

Default Header

This command examines the current circuit (i.e. the frontmost circuit window) and finds all undriven inputs and final outputs in the circuit and creates a default header in the first row of the spreadsheet area. These can then be edited manually if needed.

The following options box will be displayed:



Column Format Options

The nine buttons in the Inputs/Outputs/Expected grid allow you to select how header entries are created for input and output signals.

NOTE: "Outputs" and "Expected" both refer to the output signals, but offer different treatment of them. It is allowable to included the same signals in both places.

These options are summarized in the following table.

	Inputs	Outputs	Expected
None	No input columns ($INPUT) are included	No actual output columns ($OUTPUT) are included	No expected output columns ($EXPECTED) are included
In One Column	All inputs are included in a single column, e.g. $I A B C D	All outputs are displayed in a single column, e.g. $O A B C D	All expected outputs are checked in a single column, e.g. $E A B C D
In Separate Columns	Individual inputs or groups are shown in separate columns	Individual outputs or groups are shown in separate columns	Individual outputs or groups are checked in separate columns

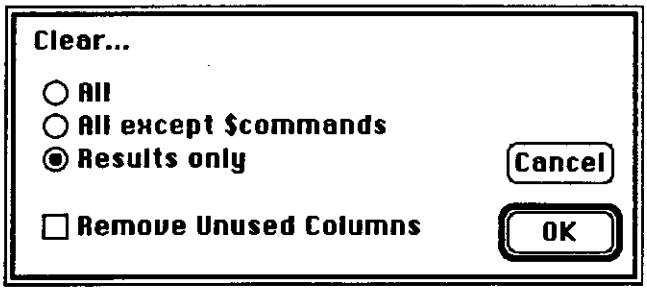
Other Default Header Options

The following table summarizes the remaining options in the Default Header box.

Show Only Signals with Port Connectors	This option is intended for use with sub-circuit blocks that have Port Connector symbols attached to all external pin connections.
Clear Existing Data	If this box is checked, all data currently in the spreadsheet area will be cleared before creating the new header. THIS CANNOT BE UNDONE!
Include Time Column	This option will insert a column at the left with a $TIME header. This will display time values as the test program executes.
Group Numbered Signals	When this option is enabled, sequentially numbered signals (e.g. FC0 FC1 FC2) will be grouped together using the "[]" grouping operator, e.g. "[FC2..0]". They can then be specified with a single hexadecimal value. The lowest numbered signal is assumed to be the least significant bit of the value. This applies to all Inputs, Outputs or Expected columns.

Clear Sheet

This command clears the entire data spreadsheet area, offering the following options:



All	All cells in the spreadsheet are cleared
All Except $Commands	Any cell starting with a "$" character will be left untouched.
Results Only	All cells that were filled in by the program will be cleared and all others left untouched.
Remove Unused Columns	If this box is checked, any columns that are completely empty will be deleted.

Monitor Mode

This command selects monitor mode operation. In this mode, any command or specification that would set up a signal value change in the circuit under test is ignored. Each time any signal specified as an output changes state, a new line is displayed in the spreadsheet area.

Selecting this command again returns to normal mode.

Copy to Timing

This button performs the "timing setup mode" conversion of test vector data to simulation events. If the DesignWorks simulator is not stopped, then simulation will proceed immediately after events are set up.

NOTES:

1) The meanings of the some of the commands in the Command column of the spreadsheet are changed in this mode. For example, an infinite repeat loop is not allowed and will be interpreted as a single iteration.

2) This command does not affect the clipboard.

Resize Columns

This command adjusts the width of all columns so that the widest data in the column just fits.

Update Display

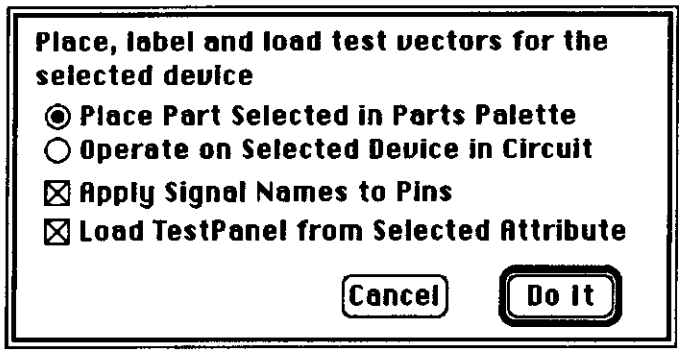
When this box is checked, the spreadsheet is updated with new data after each operation. This provides best user feedback on the progress of the test, but substantially slows down the process. This control only affects the appearance of the display. All other controls still operate interactively. This switch can be turned on and off at any time during operation to check on progress of the test.

Place and Test Device

This command provides the ability to load a device from a library and ready it for testing in a single operation. It is intended to allow a quick verification or analysis of a library part.

NOTE: Most of the parts in the Primitive libraries shipped with the Simulator contain test vectors which can be used to verify their operation or as examples for creating your own device test procedures. Simply click on the part in the Parts Palette once so that it is selected, then select this command.

This command will operate either on a part type selected in the currently displayed library in the Parts palette, or it will operate on a selected device already placed in the current circuit. It displays the following option box:


Place, label and load test vectors for the selected device
Place Part Selected in Parts Palette
Operate on Selected Device in Circuit
Apply Signal Names to Pins
Load TestPanel from Selected Attribute
Cancel
Do It

Place Part Selected in Parts Palette	If this button is selected, then the part currently selected in the Parts Palette will be placed in the center of the current schematic page. The circuit is assumed to be empty, but no checking is done.
Operate on Selected Device in Circuit	If this button is selected, then the operations selected below will be performed on the device that is currently selected in the schematic.
Apply Signal Names to Pins	If this option is checked, a visible signal name will be applied to each pin, corresponding to the name of the pin. This name can then be referred to in the test vectors.
Load TestPanel from TestVectors Attribute	If this option is selected then the spreadsheet area will be cleared and new data loaded from the TestVectors.Dev attribute field in the device.

Save Test to Device

This command causes the entire spreadsheet area to be saved to the TestVectors.Dev attribute field of the device selected in the schematic.

See the comments under the Save to Design Attr command for the format and limitations of this data.

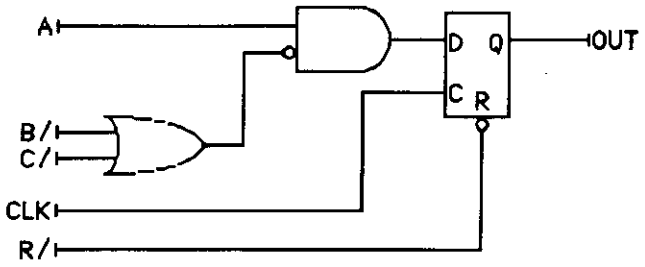
TestPanel Examples

This section provides some simple examples of TestPanel usage. For more complex examples, see the example files provided with the simulator release.

NOTE: Most of the devices in the Primitive device libraries contain stored test vectors and can be used in conjunction with the Place and Test Device command in the Options pop-up menu for example purposes. See the description of this command elsewhere in this chapter.

Testing a Circuit

In this example we will execute a set of test vectors for the following simple circuit made up of primitive devices:



This example and the test program below are provided as files with the Simulator release.

Entering the Test Program

NOTE: This file is included with the Simulator release and can be loaded using the Open Test File command in the File pop-up menu.

With the above circuit open in the frontmost schematic window, follow these steps:

- Select the TestPanel item in the Tools menu. An empty TestPanel window will appear.
- Select the Default Header command in the Options pop-up menu.
- For this example the "Include Time Column" switch was turned off, although you can optionally leave it on and shift all entries right one column.
- Click OK on the Default Header box. This will create the header entry for the first row.

• Enter the remaining cells as shown below. Note that it may be convenient to enter the first few lines manually then use the Fill Down command in the Edit pop-up menu to set the default values in the remaining rows.

$I R/	$I CLK	$I C/	$I B/	$I A	$E OUT
0	0	1	1	0	0
1	0	1	1	0	0
1	.N	1	1	0	0
1	.N	0	1	0	0
1	.N	0	0	0	0
1	.N	0	0	1	1
1	.N	1	0	1	0

We are now ready to execute the test program. You may wish to save the current entry using the Save Test File As command in the File pop-up menu.

Executing the Test Program

To execute the test program:

• Click on the Reset button. This will select the top-left cell in the spreadsheet.
• Click on the Walk button. If the simulator is Stopped, you will be prompted to start it, otherwise the program will be executed.

If all TestPanel and delay settings are the default values, this program will fail in the last step, due to a setup time problem. This will be explained in the next section.

Dealing With Delays and Setup Times

In this example, we are feeding inputs through a combinational logic array to an edge-triggered device. We must be aware that a delay must be allowed between changes at the combinational inputs and the application of the clock edge to the flip-flop.

In this case, the flip-flop is positive-edge-triggered. We have used the ".N" negative clock pulse notation since this in effect inserts a delay equal to the pulse width before the rising edge. This is because is creates a negative edge first (which is not significant to this flip-flop), then waits for the specified pulse width, then inserts the positive edge.

The default TestPanel clock pulse width is 2 units and the default delay in primitive devices is 1 unit. The last test vector in the above program causes

a logic transition to pass through two gates, for a total delay of 2 units. Thus, the input transition at the D input to the flip-flop arrives at the same time as the positive edge of the clock. The D flip-flop model always uses the old value at the input, so the new value does not reach the Q output.

There are two solutions to this problem:

- Change the TestPanel clock pulse width using the Test Options command in the Options pop-up menu. This will affect all clock pulses generated by this TestPanel window.

NOTE: These settings are not saved with the test file and will return to default values the next time the program is restarted.

- Change the test program to set the clock input explicitly and insert a delay at this line, as in the following changes to the final line:

1	.N	0	0	1	1
$DEL 5	0	1	0	1	.DC
1	1	1	0	1	0

In this case, we are replacing the .N operator with two separate test vectors with explicit 0 and 1 values for the clock and a delay of 5 in between.

TestPanel as a Pattern Generator

TestPanel can also be used as a repeating test pattern generator or even as a replacement for a device or bus interface which is not present in the circuit design.

In this case, we will use the $WAIT command to synchronize the values generated by the TestPanel with the design under test. When the SYNC signal enters the 0 state, the D0..3 lines will be set to zero. When SYNC goes to 1, the TestPanel will cycle through a Gray code sequence and then loop back to the beginning.

Creating Fixed Step Sizes

By default, the TestPanel executes each step when the simulation settles after the previous one. This is appropriate when the test program is driving the circuit and evaluating the outputs, but in this case we want a fixed step size, regardless of other simulation activity. To do this:

- Select the Test Options command in the Options pop-up menu.
- Enter the value 10 into the Default Delay box.
- Select the After Specified Delay option.
- Click OK on this box.

The TestPanel will now proceed at fixed intervals whether or not the simulation has settled.

The Test Program

The following test program performs the described function.

	$WAIT SYNC	$I [D3..0]
$REP	0	
	1	0
		1
		3
		2
		6
		7
		5
		4
		C
		D
		F
		E
		A
		B
		9
$END		8

As a variation on this, removing the 0 and 1 entries from the $WAIT column or deleting the column altogether, will cause the test program to loop continuously without synchronization.

Chapter VII - Primitive Devices

This chapter provides information on the "primitive" or built-in device types in DesignWorks. These types are intended primarily for use in creating model circuits for higher-level macro devices. Because their simulation functions are hard-coded, they occupy much less memory space than macro devices and simulate more efficiently.

IMPORTANT NOTES:

1) In primitive devices, the association of logic functions to pins on a device symbol are made by pin order. When creating primitive devices using the DevEditor tool, you must be aware of the pin order requirements for the device type you are using. Refer to the description of each type in this chapter and to Appendix D - Primitive Device Pin Summary.

2) Bus pins are not supported on primitive device types.

The following table lists the available primitives and their functions.

Primitive Type	Description	Related Type	Max. # Inputs
NOT	Inverter		1
AND	N-input AND gate	Any pin inversions	799
NAND	N-input NAND gate	Any pin inversions	799
OR	N-input OR gate	Any pin inversions	799
NOR	N-input NOR gate	Any pin inversions	799
XOR	N-input XOR gate	XOR	799
XNOR	N-input XOR gate	XNOR	799
Transmission Gate	Transmission gate		1
Buffer	Non-inverting N-bit 3-state buffer with optional common inverted enable	Buffer	400
Resistor	Digital resistor		1
Multiplexer	M*N-to-M multiplexer		256
Decoder	1-to-N line decoder		256
Adder	N-bit adder with carry in and out	Incrementer	256
Subtractor	N-bit subtractor with borrow in and out	Decrementer	256

D Flip-Flop	D-type flip-flop	optional S & R	1
JK Flip-Flop	JK flip-flop	T flip-flop, optional S & R	1
Register	N-bit edge-triggered register		256
Counter	N-bit synchronous counter	Up/down	256
Shift Register	N-bit shift register		256
One Shot	Retriggerable one shot		1
Clock	Clock oscillator		1
Binary Switch	Debounced toggle switch		1
SPST Switch	Open/closed single pole switch		1
SPDT Switch	Double throw switch		1
Logic Probe	Signal level display		1
Hex Keyboard	Hexadecimal input device		1
Hex Display	Hexadecimal digit display		1
SetupHold	Setup/Hold time checker		1
Unknown	Unknown value detector		1

The following table lists devices supported primarily for compatibility with older version of DesignWorks. We do not recommend using these in new designs.

Device	Description
Pullup	Pullup resistor, single pin
D Flip-Flop ni	D-type flip-flop (non-inv S & R)
JK Flip-Flop ni	JK flip-flop (non-inv S & R)
Glitch	Glitch detector
SimStop	Simulation halt device

Gates and Buffers

The Primitive Gates library contains the primitive gates with a built-in simulation function. The NOT, AND, NAND, OR, NOR, XOR and XNOR devices behave according to the appropriate truth tables for such devices. Any gate input which is in the "Don't Know", "High impedance" or "Conflict" state is treated as a "Don't Know". A gate with a "Don't Know" input will not necessarily produce a "Don't Know" output. For example if one input of an AND gate is low, the output will be low, regardless of the state of the other input, as in the following truth table:

A	B	OUT
0	0	0
0	1	0
0	X	0
1	0	0
1	1	1
1	X	X
X	0	0
X	1	X
X	X	X

Gate Definition

The gate types, except NOT, can be created with any number of inputs (from 0 to 799) and are defined as shown in the following table.

Function	Output is...	Output is DONT if...
AND	LOW if any input is LOW, HIGH otherwise	Some input is DONT and no input is LOW
NAND	HIGH if any input is LOW, LOW otherwise	Some input is DONT and no input is LOW
OR	HIGH if any input is HIGH, LOW otherwise	Some input is DONT and no input is HIGH
NOR	LOW if any input is HIGH, HIGH otherwise	Some input is DONT and no input is HIGH
XOR	HIGH if an odd number of HIGH inputs and no DONTs	Any input is DONT
XNOR	HIGH if an even number (or zero) of HIGH inputs and no DONTs	Any input is DONT

Gate Pin Order

The NOT type must have exactly one input and one output, in that order. All other logic gate types can have any number of inputs followed by a single output, up to the maximum DesignWorks limit of 800 pins.

NOTE: Pin order is important in all primitive devices! When creating a gate type using the DevEditor tool, the output pin must be the last item on the pin list.

Pin Inversions

The logic of any pin on any device can be inverted by placing a non-empty value in the Invert.Pin attribute field of the pin.

For example, to create the following AND gate with one inverted pin:

the following steps must be taken in the DevEditor tool:

1) Create the desired graphic symbol using the DevEditor's drawing tools.
2) Place the three pins as shown. ORDER IS IMPORTANT! All primitive devices must have a specific pin order. For gates, all inputs come first and the output pin last.
3) In the pin list, double-click on the last pin (the output pin). This will display the Pin Info palette for that pin.
4) Set the pin type to Output, and, if desired, edit the pin name.
5) Click the Enter key on the keyboard to move to the next pin and edit the other pin names, if desired, and check that they are all set to Input.
6) Click the close box in the Pin Info palette.
7) In the pin list, click once to select the input pin that is to be inverted, then select the Pin Attributes command in the DevEditor menu.
8) Select the All Fields option in the attributes box, then select the Invert.Pin field.
9) Enter the value "1" for this field, then click Done. (The actual value doesn't matter, as long as it is non-empty.)
10) Select the Sub-circuit and Part Type command in the DevEditor menu.
11) Click on the Primitive Type button, then select the AND primitive type in the pop-up menu.
12) Close the Part Type box and save the part to a library in the usual manner.

NOTES:

1) The logical inversion of the pin is completely independent of the graphical representation of the pin. I.e. using the "inverted pin" graphic in the DevEditor does not invert the pin logic in the simulator. You must set the Invert.Pin field to have this effect.

2) Inverted gate types NAND and NOR can be created by using the NAND and NOR primitive type setting or by using the AND and OR settings and inverting the output pin (or the input pins, using DeMorgan's Theorem). The two methods will produce identical simulation results. There is a slight memory overhead but no execution speed overhead to using an inverted pin.

Transmission Gate

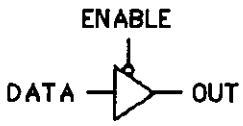
The transmission gate (X-Gate) device behaves as an electrically controlled SPST switch. When the control input is high, any level change occurring on one signal pin will be passed through to the other. Since it has no drive capability of its own it will behave differently than a typical logic device when a high-impedance or low drive level signal is applied to its signal inputs. Most other primitives, such as gates, interpret any applied input as either High, Low or Don't Know. The transmission gate, on the other hand, will pass through exactly the drive level found on its opposite pin. Thus, a high-impedance level on one pin will be transmitted as a high-impedance level on the other pin. Note that the simulation of this device may produce unpredictable results in extreme cases, such as an unbroken ring of transmission gates.

NOTE: No variations in number or order of pins are possible with the XGATE primitive type. It must have exactly one control pin and two bidirectional pins with pin order as described in Appendix D - Primitive Device Pin Summary.

Three-State Buffer

The three-state buffer has N data inputs, N data outputs, and an optional active-low enable input. If the enable input exists and is high, all outputs enter a "High-impedance" state. If the enable input doesn't exist, or is low, each output will follow the corresponding input if it is low or high, or produce a "Don't Know" level otherwise.

A single-input three-state buffer is shown in the following table:



ENABLE	DATA	OUT
0	0	0
0	1	1
1	0	Z
1	1	Z

Making Non-Inverting Buffers

The Buffer primitive type can also be used to make a non-inverting buffer, i.e. with its outputs always enabled, simply by omitting the enable input.

This can be used for the following purposes:

- To represent a non-inverting buffer or level translator in a design.
- To insert a delay in a signal path without affecting the logic of the signal.
- To create various types of open collector, open emitter or inverting buffers, when used in conjunction with different pin type and inversion settings on the outputs. NOTE: It is more efficient to use the NOT primitive type to make a simple inverter.

Resistor

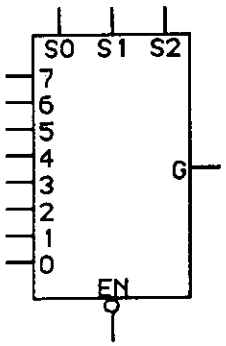
The resistor device simulates the effects of a resistor in a digital circuit. It is more general than the Pullup Resistor device and can be used as a pullup, pulldown or series resistor. Whenever a signal level change occurs on either pin of the resistor, the device converts that level into a resistive drive level (see Chapter III - Simulation for more information on drive levels). A high impedance drive on one end is transmitted as a high-impedance output to the other end. Note that DesignWorks does not simulate analog properties of devices so the resistor device does not have a resistance value in the analog sense and will not interact with capacitor symbols placed on the same line. The effect of resistance on line delay can be simulated by setting the delay of the resistor device.

Logic Devices

Multiplexer

This is a device which selects one of N data inputs and routes it to a corresponding output line. There can from 1 to 256 outputs, plus an optional enable input, as long as the total pin count does not exceed the 800 pin limit.

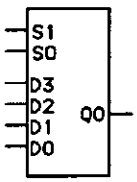
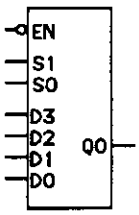
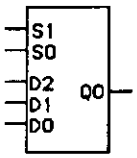
A typical 8-to-1 multiplexer obeys the following function table. X = Don't Care.



EN	S2	S1	S0	7	6	5	4	3	2	1	0	G
0	0	0	0	X	X	X	X	X	X	X	0	0
0	0	0	0	X	X	X	X	X	X	X	1	1
0	0	0	1	X	X	X	X	X	X	0	X	0
0	0	0	1	X	X	X	X	X	X	1	X	1
0	0	1	0	X	X	X	X	X	0	X	X	0
0	0	1	0	X	X	X	X	X	1	X	X	1
0	0	1	1	X	X	X	X	0	X	X	X	0
0	0	1	1	X	X	X	X	1	X	X	X	1
0	1	0	0	X	X	X	0	X	X	X	X	0
0	1	0	0	X	X	X	1	X	X	X	X	1
0	1	0	1	X	X	0	X	X	X	X	X	0
0	1	0	1	X	X	1	X	X	X	X	X	1
0	1	1	0	X	0	X	X	X	X	X	X	0
0	1	1	0	X	1	X	X	X	X	X	X	1
0	1	1	1	0	X	X	X	X	X	X	X	0
0	1	1	1	1	X	X	X	X	X	X	X	1
1	X	X	X	X	X	X	X	X	X	X	X	1

Multiplexer Pin Variations

A number of variations in multiplexer logic are possible with this primitive type, depending on which input and output pins are included. The following table summarizes the possible variations. Samples are shown with M=1 and N=2, but any combination of M and N can be used within the maximum pin limit of 800.

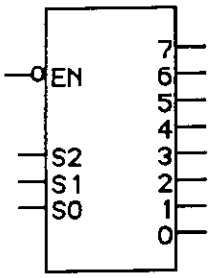
Number of Sections	Number of Inputs/Section	Number of Select Inputs	Number of Enable Inputs	Sample Symbol
M	2^N	N	0	
M	2^N	N	1	
M	2^N-1+1 .. 2^N	N	0*	

* If there are less than 2^N inputs per section there can be no enable input.

Specific pin order requirements for the multiplexer type are given in Appendix D - Primitive Device Pin Summary.

Decoder

The Decoder activates one of N outputs, depending on M select inputs, as follows (X = Don't Care):



EN	S2	S1	S0	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	0	1
0	0	1	0	1	1	1	1	1	0	1	1
0	0	1	1	1	1	1	1	0	1	1	1
0	1	0	0	1	1	1	0	1	1	1	1
0	1	0	1	1	1	0	1	1	1	1	1
0	1	1	0	1	0	1	1	1	1	1	1
0	1	1	1	0	1	1	1	1	1	1	1
1	X	X	X	1	1	1	1	1	1	1	1

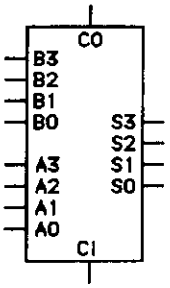
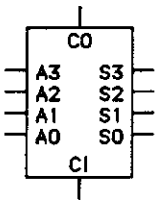
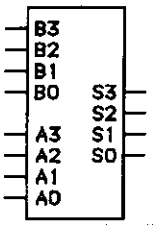
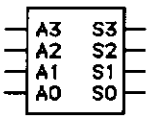
Adder/Incrementer

The N-bit Adder accepts one or two N-bit input arguments and (optionally) a 1-bit carry and outputs their N-bit sum plus an optional 1-bit carry out.

Multiple adders can be connected together by feeding the carry out from each stage to the carry in of the next more significant stage. The carry in to the least significant stage should be set to zero.

Adder Pin Variations

The adder primitive can be used in four variations, as summarized in the following table. Sample symbols are shown with 4 bit inputs, although any number of bits from 1 to 256 is permissible.

	Has B Inputs	No B Inputs
Has Carry In	S = A + B + Cin 	S = A + Cin 
No Carry In	S = A + B 	S = A + 1 

In addition, the Carry Out pin can be independently included or omitted from any of these configurations.

Refer to Appendix D - Primitive Device Pin Summary for precise pin order requirements.

Subtracter/Decrementer

The Subtracter primitive type behaves identically to the Adder type except that a subtract or decrement operation is performed, depending upon pin configuration.

D Flip-Flop

The D-type flip-flop is positive-edge triggered and obeys the following function table:

S	R	D	Clock	Q	Q/
0	0	X	X	1	1
0	1	X	X	1	0
1	0	X	X	0	1
1	1	0	Rises	0	1
1	1	1	Rises	1	0
Rises	Rises	X	X	X	X

In the above table X on the input side means "Don't Care" and on the output side means "Don't Know".

Flip-Flop Setup and Hold Times

None of the DesignWorks primitive types explicitly implement variable setup and hold times. However, all edge-triggered primitives have an effective setup time of 1 unit since they always use the input signal value existing before the current step. I.e. if the data input changes at the same time as the clock, the old data value will be used to determine the new output value. This effective setup time can be modified by specifying input pin delays on either the data or clock pins.

Checking for setup and hold violations can be done in one of two ways:

- Attaching a SetupHold primitive device to the inputs and output of the device under test. The combined device could be made into a sub-circuit if desired to combine it into a single symbol.
- Using the Trigger capability of the simulator to watch for value changes within a set amount of time.

Flip-Flop Initialization

Note that when a flip-flop is first placed in the schematic, it is in an unknown state and must be correctly initialized before it will produce predictable outputs. This can be done in the following ways:

- Adding circuitry to force an explicit reset.
- Using the Clear Unknowns command in the Edit menu to force an initial state before starting the simulation.
- Specifying an initial output value for both the Q and Q/ outputs in their respective Initial.Pin attributes. This will be applied every time a Clear Simulation command is executed.

D Flip-Flop Optional Pins

The D Flip-Flop primitive type has the following optional pins:

- The Q/ (Not-Q) output can always be omitted.
- The Set input alone or both the Set and Clear inputs can be omitted.

Refer to Appendix D - Primitive Device Pin Summary for specific pin order information.

D Latch

The D Latch primitive type is identical to the D Flip-Flop in function and pin specifications except that it is level-triggered instead of edge-triggered. I.e. The Q and Q/ outputs will follow the level of the D input as long as C is high.

JK Flip-Flop

The JK flip-flop is negative-edge triggered and obeys the following function table:

S	R	J	K	Clock	Old Q	New Q	New Q/
0	0	X	X	X	X	1	1
0	1	X	X	X	X	1	0
1	0	X	X	X	X	0	1
1	1	0	0	falls	0	0	1
1	1	0	0	falls	1	1	0
1	1	0	1	falls	X	0	1
1	1	1	0	falls	X	1	0
1	1	1	1	falls	0	1	0
1	1	1	1	falls	1	0	1
rises	rises	X	X	X	X	X	X

In the above table X on the input side means "Don't Care" and on the output side means "Don't Know".

If any inputs are in an unknown state, the simulator will determine the output state where possible, or else set it to "Don't Know".

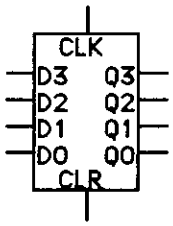
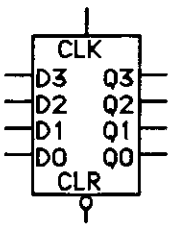
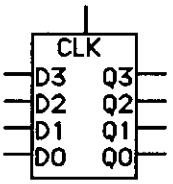
See the notes under D Flip-Flop, above, on setup and hold times and initialization.

Register

This device implements an N-bit, positive-edge-triggered register with common clock and an optional active-high clear inputs.

See the comments on Setup and Hold times and initialization in the D Flip-Flop section, earlier in this chapter.

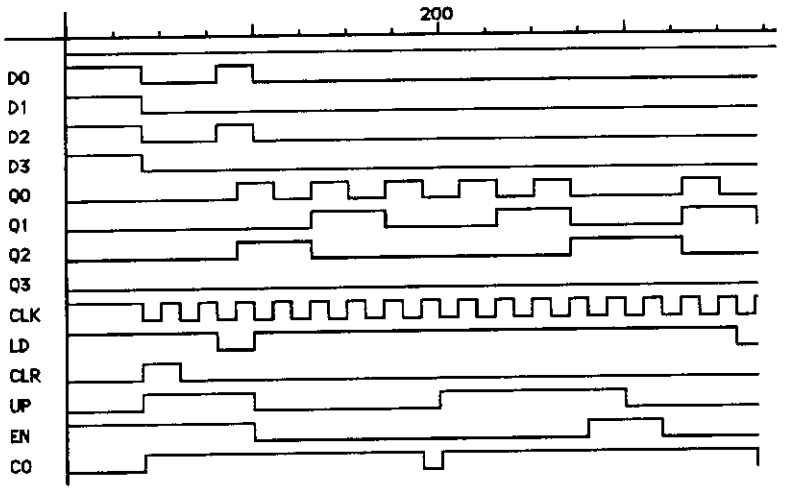
The following table illustrates some pin variations available for the Register primitive type.

4-bit register with active-high clear	
4-bit register with active-low clear (using pin inversion)	
4-bit register without clear	

Counter

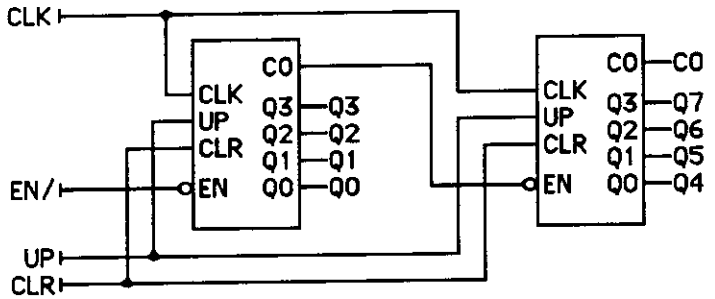
This device implements a N-bit, presettable, synchronous, positive-edge-triggered up/down counter with active-low enable. The load data inputs and most of the control inputs can be omitted for simplified versions.

The following timing diagram shows a typical count cycle. Note that the CO (Carry Out) output goes low when the count reaches 2^N-1 (when counting up) or 0 (when counting down) and rises again on the next count. This can be used to cascade multiple counters together, as shown. The CLR input clears the counter asynchronously (i.e. regardless of the state of the clock). The Count/Load input, when low, causes the data from the N data inputs to be passed to the outputs on the rising edge of the next clock. The Enable input disables counting when high, but has no effect on loading.



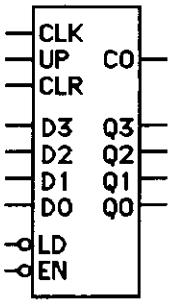
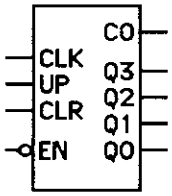
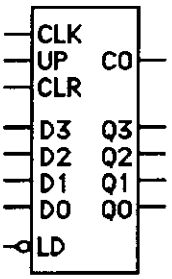
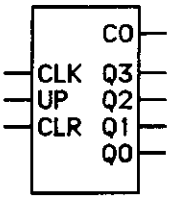
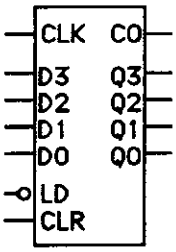
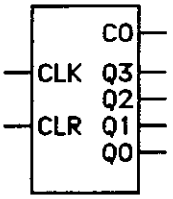
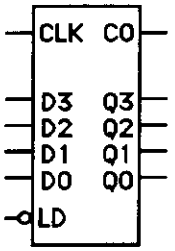
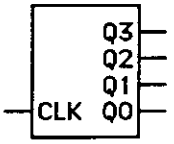
Cascading Multiple Counters

Counter primitives with the optional Enable and Carry Out pins can be cascaded to form larger synchronous counters as follows:



Counter Pin Variations

The following table summarizes the possible pin usage variations for the counter primitive type. The sample are shown with N=4, although the number of bits can be anywhere in the range 1 to 256.

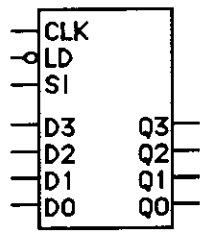
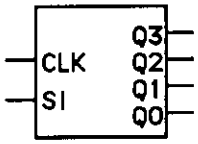
Optional Inputs	Including Load Inputs	Excluding Load Inputs
CLR, UP/DN, ENABLE		
CLR, UP/DN		
CLR		
none		

NOTE: COUT can be independently included or omitted in any of the above variations.

Shift Register

The shift register is an N-bit, positive-edge-triggered device with serial or optional parallel load. When the Shift/Load input is low, data from the N parallel data input lines is transferred to the outputs on the rising edge of the next clock. When Shift/Load is high, the next rising clock edge causes the value at the Shift In input (SI) becomes the new value for output Q0, Q0 shifts to Q1, Q1 to Q2, etc., and the old value at the most significant output is lost.

The following table shows the shift register primitive with and without parallel inputs.

With Parallel Load	Without Parallel Load
	

NOTE: The Shift Register primitive cannot be created without data outputs (i.e. a parallel-in serial-out register) because the flip-flop values are stored on the output pins. Primitive devices have no internal state storage. See more comments on this in Chapter III - Simulation.

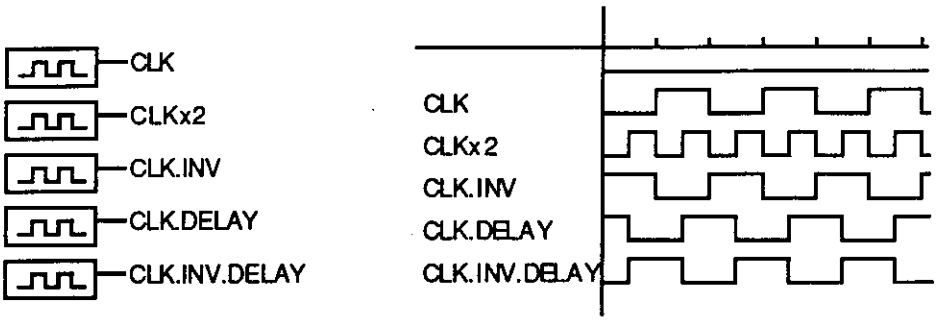
Clock

The clock oscillator is used to generate a repeating signal to activate other devices. When it is first created, the clock output pin will be low, then after a delay time called the "Low Time" it will change to the high state. After a further delay called the "High Time" the signal will revert to low, and the cycle will repeat. The low and high times are initially set to 10, but can be modified using the Simulation Params item in the "Simulation" menu. Any number of Clocks may exist at once with independent delay times.

Creating Synchronized or Offset Clocks

When the Clear Simulation command is selected (or the Restart button on the Timing palette pressed), all clocks in the design are restarted. Clock outputs will be set to the low state and the timer for the low period will be restarted. Clock high and low times, combined with pin inversion and pin

delay settings can be used to precisely determine the relationship between two clock outputs. The following circuit example summarizes these options.



Signal	Low Time	High Time	Invert.Pin	Pin Delay
CLK	10	10		0
CLKx2	5	5		0
CLK.INV	10	10	1	0
CLK.DELAY	10	10		5
CLK.INV.DELAY	10	10	1	5

Setting Clock Values

To set the high and low times for a clock, first select the device in question (by setting the cursor to "Point" mode and clicking the mouse button inside the device symbol), then choose the **Simulation Params** item in the Simulation menu.

You will be presented with a dialog box with buttons for increasing or decreasing the high and low values. The minimum for either value is 1 and the maximum is 32767.

See more information on the Simulation Params command in Chapter IX - Menu Reference.

One Shot

The one shot is used to generate an output pulse of a fixed length when it is triggered by the rising edge of the trigger input. Two parameters can be set for a one shot: the delay from the rising edge of the input to the start of the output pulse, and the duration of the pulse. The delay and duration times are initially set to 1 and 10, respectively, but can be modified using the Simulation Params item in the "Simulation" menu.

The One Shot device is retriggerable, meaning that the output pulse will not end until "duration" time units has passed since the last trigger input. Repeating the trigger input can cause the output pulse to be extended indefinitely.

Setting One Shot Values

To set the delay and duration times for a one shot, first select the device in question (by setting the cursor to "Point" mode and clicking the mouse button inside the device symbol), then choose the Simulation Params command in the Simulation menu.

Refer to Chapter IX - Menu Reference for more information on the Simulation Params command.

I/O Simulation Pseudo-Devices

Binary Switch

The Binary Switch device provides a means for setting a signal to a low or high level. When a switch is first created its output is at a low level. Pointing at the switch with the cursor in "Point" mode and clicking the mouse button causes the switch arm to move and the output to change to the opposite state. Any number of device inputs can be driven by a switch output. A switch has no delay characteristic since it has no inputs.

SPST Switch

The SPST switch device simulates the actions of a simple open/closed switch in a digital circuit. When a switch is first created it is open and both connections present a high impedance logic level. Clicking on the switch with the cursor in "Point" mode causes the switch arm to close and the switch to "conduct". In terms of the digital simulation, this means that whatever logic level is present on each pin is transmitted to the other one.

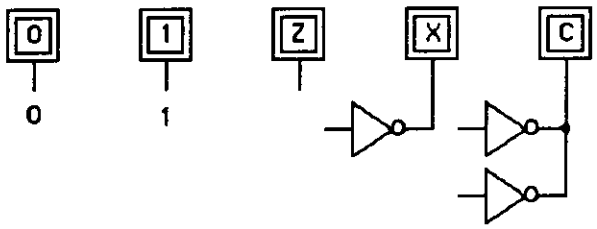
An SPST switch has a default delay of zero but this can be set to any value from 0 to 32767 using the Simulation Params command. The Shift key must be pressed on the keyboard in order to select a switch device.

SPDT Switch

The SPDT switch device operates in essentially the same manner as the SPST switch described above, except that it is always conducts between the single pin on one side and one of the two pins on the other. As with the other two switch types, clicking on it with the Point cursor changes the position of the contact.

Logic Probe

The logic probe is a device for displaying the level present on any signal line. When the probe is first created, its input is unconnected and therefore in the "High-impedance" state, which will be displayed as a "Z". When the input pin is connected to another signal, the displayed character will change to reflect the new signal's current state. Any further changes in the signal state will be shown on the probe. Possible displayed values are 0 (low), 1 (high), X (Don't Know), Z (High impedance) or C (Conflict).



Hex Keyboard

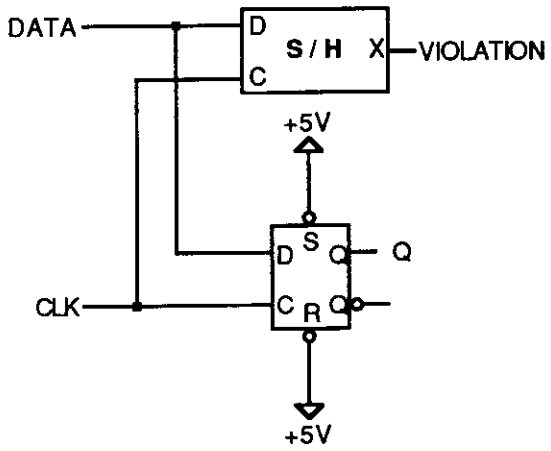
The hex keyboard outputs the binary equivalent of a hexadecimal digit on four binary lines. A "key" is pressed by positioning the tip of the arrow cursor in the desired key number and clicking the mouse button. The binary data on the output lines will change to reflect the new value and remain until the next key is pressed. On a fifth line, it outputs a signal which goes high momentarily and then low again when a key is pressed.

Hex Display

The hex display shows the hexadecimal equivalent of its four binary inputs. If any of the inputs is unknown, high-impedance or conflict, then an X will be displayed.

SetupHold

Setup and hold times can be checked by attaching a "SetupHold" primitive device to the inputs of the clocked device to be checked, as follows:



Each time the "D" input changes, the SetupHold device marks the current time. On the next rising edge of the "C" input, the current time is compared to the last transition on the "D" input. If the difference is less than or equal to the Setup time specified in the Simulation Params box, the output switches to a Don't Know state and remains in this state until a non-violating input transition occurs. Similarly, for each clock rising edge, the time is marked and compared to the time at the next transition on the "D" input. If this difference is less than the Hold time given, the output is set to Don't Know.

The SetupHold device puts out a highZ value until a setup or hold violation occurs when it switches to a don't know state. Thus, the output of the SetupHold device can be paralleled with the flip-flop so that the output line will enter a conflict state when an error occurs.

Unknown Detector

The Unknown Detector device detects any unknown value (i.e. not 0 or 1) on its input and puts out a high value after the delay time specified in the Simulation Params box. When the input returns to a 0 or 1 state, the output becomes zero. This can be used in conjunction with the SimStop device to detect invalid conditions in a circuit and stop the simulator.

RAM Devices

The RAM primitive device supports the direct simulation of static Random Access Memory devices in a variety of configurations. A number of common RAM types are provided in the Primitive RAM library. In addition, the RAM tool allows you to create custom RAM devices with a variety of pin options.

NOTE: To create or simulate RAM devices, the RAM tool must be installed in the Tools folder.

Description of the RAM Primitive

The following table summarizes the options available in the RAM primitive type.

Chip Enables	0, 1, 2 or 3 active-low chip enables. If any chip enable input is high, all read and write functions are disabled. In no enable is provided, the device will always be enabled.
Write Enable	The active-low Write Enable pin is not optional. A low level on this pin causes the data present at the Data In lines to be written to the location selected by the address lines.
Output Enable Pin	This active-low pin controls output enable but does not affect writing.
Data In/Out	The data input and output lines can either be separate or combined into a single I/O bus.
Three-state outputs	If the input and output lines are combined, or if three-state outputs are specified, the outputs enter a high-impedance state if Write Enable or any Chip Enable are high. If three-state outputs are not specified, data outputs will be high when disabled.
Single-word Simulation	If this option is selected, only a single word of real memory will be allocated for simulation purposes, i.e. the address inputs will be ignored. This allows logic testing of a circuit containing a large RAM device without consuming large amounts of program memory.
Common I/O	This option specifies that a single I/O pin will be used per data bit, rather than separate data in and data out lines. In this case, three-state outputs are assumed and outputs will be disabled when writing.

Don't Know Input Handling

If any combination of Don't Know values on the control inputs could cause a write, then the selected memory location is invalidated (i.e. the location will contain Don't Know values). If the address inputs also have Don't Know values, then the entire device will be invalidated.

RAM Pin Options

The normal pin delay (using the Delay.Pin attribute field) and pin inversion (using the Invert.Pin attribute field) options can be used with RAM devices.

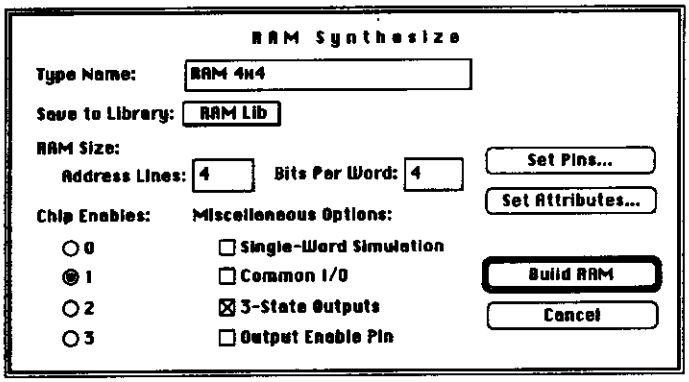
See Chapter III - Simulation for more information on pin delay and inversion.

Creating Custom RAM Devices

The RAM tool allows you to create RAM part entries in a library with any combination of optional features.

NOTE: The RAM tool will require you to select a library to which the new device definition will be saved. You may wish to create a library for this purpose (using the pop-up menu in the parts palette) before proceeding.

Select the RAM item in the Tools menu. The following parameters box will appear:



The following table summarizes the controls in this box.

Type Name	The device will be saved to the library under this name.
Save To Library	This pop-up menu allows you to select an open library to save the device to.
Address Lines	This specifies the number of address lines for the device, in the range 1 to 30
Bits Per Word	This specifies the number of bits (1 to 256) per memory location, which will also be the number of data input pins and data output pins.
Chip Enables	These buttons allow you to select 0, 1, 2 or 3 chip enable inputs.
Single-word Simulation	This check box enables the single-word simulation option, described in the previous section.
Common I/O	This option specifies that a single I/O pin will be used per data bit, rather than separate data in and data out lines.
Output Enable Pin	This will add an additional control pin which controls output enable but does not affect writing.
Set Pins	This button will display a box allowing pin names and default settings to be edited for each pin.
Set Attributes	This button displays the standard attribute edit box for part attributes.
Build RAM	When this button is clicked, a device having the selected features will be created and written to the selected library.

RAM Size Limits

RAM devices must meet all of the following limits:

- 30 address line inputs.
- 256 bits per word.
- Total RAM memory space < 2^{31} bytes (even if single word simulation is used).
- Sufficient program memory free to allocate a block twice the size of the RAM memory space, unless single word simulation is used.

Editing RAM Devices

There is no way to re-enter the RAM parameters box with an existing device. Once a RAM device definition has been created in a library, limited changes can be made by editing it using the DevEditor tool.

> IMPORTANT NOTE: The RAM device definition contains structure information that cannot be edited after the device is created. Adding or deleting any pins using the DevEditor will invalidate the device definition and render it useless.

The DevEditor can be used to make the following changes to a RAM device, if desired:

- any graphical changes to the symbol.
- pin name, visible pin number or pin attribute changes (including pin delay and inversion).
- limited pin type changes (e.g. changing to open collector).
- part attribute changes.

Chapter VIII - Programmable Devices

This chapter provides details on creating and using PROM (Programmable Read Only Memory) and PLA (Programmable Logic Array) devices with user-specified data. The use of the FromABEL tool to read external data files and simulate industry Programmable Logic Device types is covered later in this chapter.

The PROM and PLA Primitive Types

DesignWorks supports the direct simulation of PROM and PLA devices as primitives. This means that these devices can be efficiently represented as a single simulation device, rather than having to generate a netlist of equivalent logic devices.

The PROM and PLA devices represent "raw" memory or PLA (AND-OR) arrays. The primitive device models do not include capability for registers, feedback, three-state buffers or other device features. In order to model these features in industry PLD and PROM types, the FromABEL tool automatically generates a sub-circuit model made up of a raw PLA device, plus other primitive registers, buffers, etc. as needed.

PROM Device Format

For the purposes of simulation in DesignWorks, a PROM (Programmable Read Only Memory) is defined as a device having N inputs (from 1 to 30) and M outputs (from 1 to 256), and having 2^N storage locations, each containing M bits. Each different input combination selects one of the storage locations, the contents of which appear on the output lines. The number of storage locations required doubles for each input bit added, so PROM organization is only practical for a relatively small number of inputs. The advantage of the PROM is that any arbitrary Boolean function can be represented simply by storing the truth table for the function in the appropriate storage locations.

PROM Size Limits

PROM devices must fall within all of the following limits:

- 30 address line inputs.
- 256 bits per word.
- Total PROM memory space < 2^{31} bytes.
- Sufficient program memory free to allocate a block twice the size of the PROM memory space.

PLA Device Format

In DesignWorks, a PLA (Programmable Logic Array) consists (in effect) of a group of AND gates feeding into a single OR (active high) or NOR (active low) gate for each output bit. Each AND-gate input is connected to either an input bit, the inverse of an input bit, or constant high. By selectively making these input connections it is possible to determine which input combinations will produce 0's or 1's in the outputs. PLAs are actually represented internally in a compact binary format, not as a netlist of AND and OR gates.

The input connections required to implement simple logic functions can generally be determined "by eye" for simple cases, whereas more complex logic must be reduced using Karnaugh maps, the Quine-McClusky method or other more advanced design techniques. These methods are discussed in numerous circuit design textbooks and will not be covered here. DesignWorks has the capability of reading device data produced by external logic compiler programs such as MacABEL™.

PLA Size Limits

PLA devices must fall within the following limits:

- Number of Inputs + Number of Outputs <= 800.
- Number of product terms per output <= 65535.

Complex Programmable Logic Devices

The term Programmable Logic Device will be used here to refer to a real programmable device which consists of one or more AND-OR planes plus associated registers, buffers, feedback paths, etc. There is no PLD "primitive" device in DesignWorks. Some very simple PLDs can be directly simulated with a single PLA primitive type, for example a PAL10L8 type device. However, most PLDs are simulated by creating a sub-circuit containing one or more PLA primitive devices plus the other

required logic and wiring. In most cases, these sub-circuits are created automatically by the FromABEL tool, which is described later in this chapter.

Creating PROMs, PLAs and PLDs for Simulation

No manual method of entering PROM or PLA data is provided in DesignWorks. The internal data for these devices must be read from a text file in a specific format. These files are intended to be machine-generated using a program such as MacABEL™, or some other compiler package.

Two general methods are available for creating these devices, depending on type:

- PROM devices are created using the PromPLA tool. This tool allows you to manually specify the format of the device (i.e. number of inputs and outputs) and then to read the internal data from a hex file in standard format.
- PLA/PLD devices are created using the FromABEL tool. All information about the device, including the number and exact functions of inputs and outputs, is described in a text file. This file is known as a DesignWorks Link File (or "dwill", after the ".dwl" file name extension used). These files are normally created using the MacABEL programmable logic compiler, although other external means can be used to produce them.

The formats of these files are described in technical notes provided with this manual.

Creating PROM Devices

The standard hex file format used for loading PROM data does not include any information on device structure, but only a raw listing of binary data. For this reason, the PromPLA tool includes the ability to specify the parameters of a PROM device manually. The internal data must always be read from a file.

Generating the Data File

The hex files used for PROM loading are in the commonly-used Intel MDS hex format. This format can be generated by the MacABEL logic compiler and by many microprocessor assemblers and compilers. The file format is described in a technical note provided with this manual.

The PromPLA Tool

The PromPLA tool allows you to manually specify the format of the device (i.e. number of inputs and outputs) and then to read the internal data from a hex file in standard format.

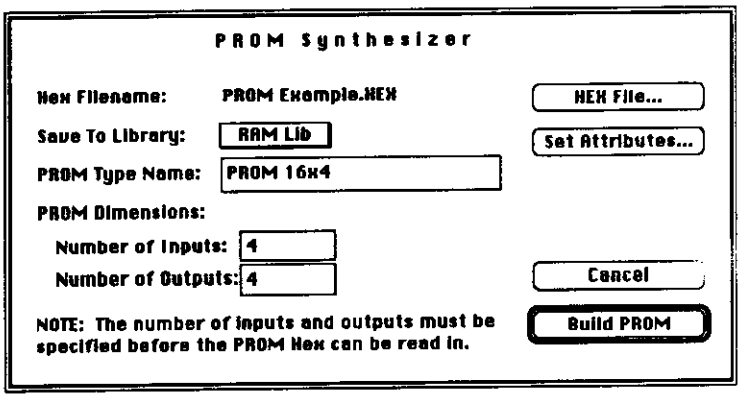
Installation

See Chapter II - Getting Started for general information on installing tools.

Starting PromPLA

NOTE: The PromPLA tool will require you to select a library to which the new device definition will be saved. You may wish to create a library for this purpose (using the pop-up menu in the parts palette) before proceeding.

To invoke PromPLA, select it from the Tools menu. This opens the following parameter box:



The following table summarizes the options in this box.

Hex Filename	This shows the currently-selected input file. This is set using the HEX File button.
Save To Library	This pop-up menu allows you to select the destination library for the completed part.
PROM Type Name	The name of the part as it will appear in the parts palette and on the symbol.
Number of Inputs	The number of address inputs.
Number of Outputs	The number of data outputs, i.e. the number of bits per storage location.

HEX File	This button allows you to select the hex data file. By default, only files with the extension ".hex" are shown. A button on this file box allows you to display all text files. The format of this file is described in a technical note supplied with this manual.
Set Attributes	This button displays the standard attribute edit box allowing you to add default attribute values.

Editing PROM Devices

There is no way to re-enter the PROM parameters box with an existing device. Once a PROM device definition has been created in a library, limited changes can be made by editing it using the DevEditor tool.

IMPORTANT NOTE: The PROM device definition contains structure information that cannot be edited after the device is created. Adding or deleting any pins using the DevEditor will invalidate the device definition and render it useless.

The DevEditor can be used to make the following changes to a PROM device, if desired:

- any graphical changes to the symbol.
- pin name, visible pin number or pin attribute changes (including pin delay and inversion).
- limited pin type changes (e.g. changing to open collector).
- part attribute changes.

Reloading PROM Devices

A PROM can be reloaded from a new hex file after it is placed on the schematic, by the following procedure:

- Select the device on the schematic.
- Select the PromPLA item in the Tools menu.
- Click on the Reload button.
- Select the desired new hex file.

The updated device can be saved back to a library, if desired, using the Schematic tool's Save to Library command.

Viewing PROM or PLA Contents

The contents of an existing PROM device on a schematic can be viewed by selecting the device on the schematic, then selecting the PromPLA item in the Tools menu.

Creating Programmable Logic Devices

The FromABEL Tool

The FromABEL tool allows you to synthesize Programmable Logic Devices (PLDs) designed with MacABEL™ 1.1 or newer software. This is used to supplement the MacABEL software by allowing you to simulate PLDs with the rest of your design in the DesignWorks environment.

Installation

See Chapter II - Getting Started for general information on installing tools.

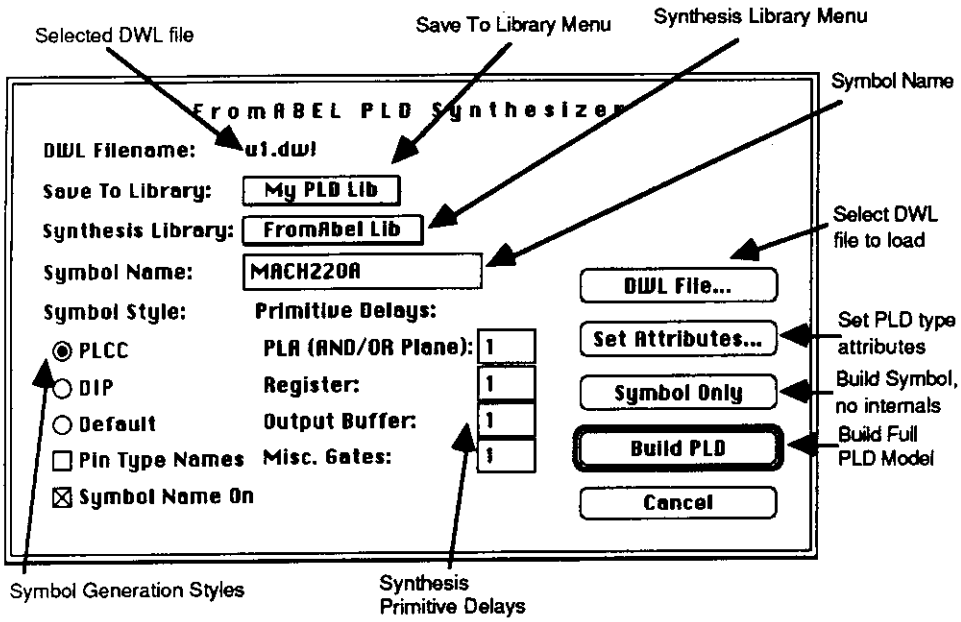
Generating the Data File

The MacABEL software provides an option to allow you to produce a "DesignWorks Link File" (*.dwl). These files describe the structure and logic of the MacABEL PLDs. Refer to MacABEL's Simulate Equations, Simulate Optimized and Simulate JEDEC commands.

Starting FromABEL

To invoke FromABEL, select it from the Tools menu. This opens the parameter dialog described in the following section. Proceed by selecting the desired ".dwl" file and setting any desired synthesis options. Then initiate the FromABEL building process. FromABEL will display a synthesis progress box showing details of the building process.

The FromABEL Parameters Dialog



The following table summarizes the features available in the FromABEL Parameters Dialog.

DWL Filename	This displays the last DesignWorks Link file selected and successfully loaded for synthesis operations. All structural & logical information will be extracted from this file.
Save To Library Menu	This pop-up menu displays the list of DesignWorks libraries currently open. Choose the destination library for the newly built PLD from this list.
Synthesis Library Menu	This pop-up menu displays the list of currently open DesignWorks libraries. Choose the source library for PLD synthesis primitives from this list (usually "FromABEL Lib", supplied with the package).
Symbol Name	Set to the value specified in the last DWL file loaded. Can be edited to a new name. This is used as the library name and the symbol name, if enabled (See "Symbol Name On" below).

PLCC Symbol Style	Creates a symbol with an equal number of pins on each of the four sides. Pin #1 is placed at the top-middle of the symbol and the other pins are added sequentially from this point in a counterclockwise fashion. Note that all unused pins will appear on the symbol. This is useful for large parts where the default method would create very tall symbols.
DIP Symbol Style	Creates a symbol where the number of pins are split equally on the left and right sides. Pin #1 starts in the upper-left and proceeds sequentially down the left side and up the right side. Note that all unused pins will appear on the symbol.
Default Symbol Style	Creates symbol using only the pins declared in your MacABEL source file and places any pin that is an OUTPUT-only signal on the right side and all others on the left side. Pin ordering is dependent on their declarations and spelling in the MacABEL source file.
Pin Type Names	Enabling this option will cause FromABEL to ignore your MacABEL pin names and use standard names for pins based on their I/O types: I - Dedicated input pin, O - Dedicated output pin, IO - Bidirectional pin, GND - Ground pin, VCC - Power pin, NC - Unused pin.
Symbol Name On	Enabling this option will cause FromABEL to generate a symbol with name that is entered in the "Symbol Name" text field, described above.
Primitive Delays	The four primitive delay text fields require an integer value (0-32767). This will allow you to obtain a functional model that more closely models the real device. The default delay for synthesis primitives is one (1). Avoid using values of zero (0), especially where feedback is concerned.
DWL File	Brings up a select file dialog box allowing you to choose the DesignWorks Link file which contains the PLD description.
Set Attributes	Brings up the standard DesignWorks attribute editing dialog. You edit or add attributes that will be specific to this PLD type. See the DesignWorks/Schematic Reference Manual for information on attributes and usage of this dialog.

Symbol Only	This causes FromABEL to ignore the structural internals of your PLD and produce a symbol only for schematic purposes, i.e. no simulation.
Build PLD	This will cause FromABEL to build the entire PLD including the internal circuit for simulation from the DWL file selected and the settings in this dialog.
Cancel Button	Quit FromABEL without doing anything.

The FromABEL Synthesis Process

The FromABEL tool makes use of many DesignWorks resources. It uses the Schematic, LibIO, DevEditor and PromPLA tools via the MEDA interface in creating PLDs. This is done invisibly and in memory, i.e. it doesn't write out any circuit files to disk. The FromABEL synthesis process is as follows:

- Create the main PLA(s) using the PromPLA tool.
- Load the synthesis elements from the library.
- Calculate the pre-place & routing parameters of the internal circuit.
- Place the main PLA(s) in the internal circuit.
- Build and attach the input nodes to the internal circuit.
- Build and attach the output nodes to the internal circuit.
- Route any global signals such as clocks, output enables, etc.
- Build new type using the DevEditor.
- Attach new internal circuit to the new type.
- Auto generate a symbol for the PLD with user settings
- Add necessary DesignWorks attributes to the new PLD for later schematic and netlist purposes.
- Save new PLD type record to the appropriate library.

This process can take anywhere from 5 seconds to a couple of minutes depending on the PLD's complexity and size.

Limited or Unsupported PLDs

The FromABEL tool doesn't guarantee to correctly build or support the following PLDs that MacABEL 4.20 supports:

F507	F839	P16Z8	P22IP6	P23S8
P241	P330	P331	P332	P448

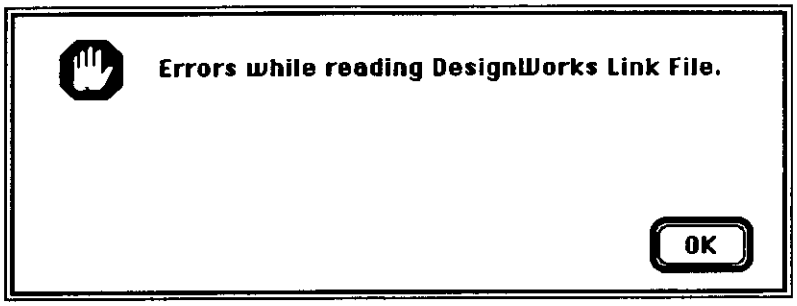
IMPORTANT: If the PLD you are using is NOT supported by FromABEL it may still synthesize without any errors. In this case, the functional simulation results are not guaranteed to be accurate.

FromABEL may correctly build some variations of the E5AC312, P241, P330, P331, and P448 parts.

Limited support of the P16RA8 and P20RA10 is available in registered mode only. Similarly, the P32VX10 has limited support for its bypass mode.

FromABEL Errors

The FromABEL errors box is used to display status messages about why FromABEL failed to create your PLD:



The most common errors are related to not having enough memory to build the PLD. Other errors result from FromABEL not finding the resources it requires to build your PLD, which again can be memory related. The other errors that may be displayed in this box include:

Errors while reading DesignWorks Link File	This will occur generally when there isn't enough memory to load the data in from the ".dwl" file. Alternatively, this can come up if the file has any been corrupted or doesn't match the version support by your copy of FromABEL.
Error trying to open link file	This will occur if the file is busy, i.e. opened in some other application.
Unsupported PLD or function, CODE = 6000	This error message will come if you are using a PLD with such a configuration that is unsupported by the internal circuit building phase of FromABEL. Take note of the "CODE" number that appears and call for MacABEL Technical Support. This error generally occurs when using a PLD that isn't supported by FromABEL.

Chapter IX - Menu Reference

The DesignWorks Simulator is made up of the Simulator core tool, called Sim, and a number of optional tools. Since each tool has its own menu commands, the menu items that are available and enabled depend upon which tools are installed and which window is currently active.

Note re TestPanel Menus

The TestPanel is a special case because it is a "palette" and always floats above the schematic and timing windows. For this reason is has its own pop-up menus in the TestPanel window and does not use any of the menu commands in the menu bar at the top of the screen. The menus at the top of the screen will continue to affect the current document window and do not affect the TestPanel.

TestPanel menus are described in Chapter VI - The TestPanel Tool.

File Menu Commands

The following commands are all associated with the Timing tool. The TestPanel has its own File pop-up menu.

Open Timing Text

This command clears the timing diagram, then opens the selected timing text data file and pastes the data onto the diagram. This is equivalent to selecting the Clear Simulation command, then using the Paste command to place the file data at time zero. See the rules for the Paste command below.

NOTE: This command does not display or remove any traces in the timing diagram. It only reads signal event data and associates it with matching traces. If any traces are named in the file that are not currently displayed, you will be warned and that data will be skipped.

Save Timing Text As

This command saves all the displayed data in the timing diagram to a text file. This file can be used for external purposes, or can be reloaded as a setup for a new simulation using the Open Timing Text command.

> *See Appendix A - Timing Text Data Format for a description of the file format.*

Print Timing

This command prints the contents of the timing window using the current page setup. The current display will be divided into as many pages a required.

Page Setup

This command determines the page setup for the Print Timing command. This can be different than the setup for the schematic diagram.

Edit Menu Commands

The following commands are all associated with the Timing tool. The TestPanel has its own Edit pop-up menu which is described later in this chapter under TestPanel Menus.

Undo

This command undoes the last editing operation in the Timing window. Unlike the Schematic tool, Timing supports only a single Undo and no Redo operation.

Cut

This command copies to the clipboard any signal change events on selected signals in the selected time interval and clears the selected interval. The data is stored in both picture form and text data form (i.e. TestPanel command

format). Events after the selected interval are not moved forward. The Delete Time command can be used to do this.

NOTES:

1) If you wish to paste a timing picture into a word processing package, it may be necessary to paste it first into a drawing program to select the picture data on the clipboard. A word processing package will normally take the text data from the clipboard by default.

2) You cannot modify timing data that is older than the current simulation time.

See Appendix A - Timing Text Data Format for information on the text format used to store timing data on the clipboard.

Copy

The Copy command copies the selected timing data to the clipboard in picture and text format. See the notes under the Cut command, above.

Note that Copy can be used on a selection to the left of (older than) the current simulation time since it does not modify the selected data.

Paste

The Paste command pastes the text timing data from the clipboard onto the selected area of the timing diagram. The following rules are used for matching the data on the clipboard with the selected interval on the timing diagram:

- Data is always pasted by name, i.e. the name of a signal in the clipboard data will be matched the same-named signal in the timing diagram. Neither the order of the signals in the clipboard data or the selected status of traces on the timing diagram is significant. To paste data from one signal to a signal with a different name, it is necessary to paste it first into the TestPanel or into a text editor, modify the names, then paste it back.
- The Paste operation affects only signals named in the clipboard data, regardless of the selection on the timing diagram.
- The Paste operation will not locate signals in the schematic that are not currently displayed in the timing diagram. No new traces will be added by this operation.

• If the time interval selected on the timing diagram is non-zero width then the selected interval is deleted and all later events on pasted signals are moved forward. A time interval equal to the width of the clipboard data is then inserted and the new data pasted into this interval.

See more information on timing diagram editing in Chapter V - The Timing Diagram.

Clear

The Clear command clears any signal change events in the selected area on the timing diagram.

NOTE: This may affect the value of a signal after the selected time interval. E.g. removing a 0-to-1 transition from a signal will leave the signal in a zero state indefinitely.

Duplicate

This command inserts a duplicate of all selected signal data on the timing diagram after the selected interval. I.e. the selected data is copied to a temporary location, then the selection point is moved to the end of the selected interval and the copied data is inserted at this point. All signal changes after the duplicate data are moved back in time by the width of the original selection.

Point

This places the timing diagram in Point cursor mode. This mode can be used to select data for clipboard operations.

Draw Sig

This command places the timing diagram in Draw Signal cursor mode. This mode allows drawing and editing signal change events.

Timing diagram editing is discussed in Chapter V - The Timing Diagram.

Select All

This command selects all traces and the entire time interval of the timing display.

Insert Time

This command inserts a blank time interval in the selected traces. The new interval is inserted in front of the selected interval and is of the same width as the selected interval.

Delete Time

This command deletes the selected time interval from the selected traces and moves all later data ahead by the width of the interval.

Simulation Menu Commands

The Simulation menu contains commands for controlling simulation and timing diagram functions, and setting device simulation characteristics. If the Timing tool is not installed, the commands affecting the timing window will not appear.

Core Simulation Commands

Speed

The Speed sub-menu is used to control the simulation speed, i.e. the amount of delay inserted between simulation steps. Simulation speed can be set individually for each open circuit.

Stop

This command stops the simulation immediately. No simulation processing is done when the simulator is in this state.

Run

This command tells the simulation to proceed as fast as possible.

Other Simulation Speeds

The intermediate speed settings between Stop and Run insert various amounts of delay between executing successive simulation time steps. These can be used to slow the simulation progress for convenient observation.

Single Step

This command simulates one time step. To perform the single step, the simulator looks at the time value associated with the next signal change event in the queue, simulates the effect of that and all following events scheduled at the same time, then returns to the stopped state. The actual time value of a single step depends on the nature of the circuit.

Simulation Params

The Simulation Params command is a general method of setting device and pin delays and options. If no devices or pins are selected in the circuit then Simulation Params will be disabled.

The type of box that is displayed will depend upon the types of devices selected, as described in the following:

Selection	Params Box	Notes
A single CLOCK device	Clock Params Box	Only one clock device can be set at a time
A single SETUPHOLD device	SetupHold Params Box	Only one SetupHold device can be set at a time.
A single ONESHOT device	One Shot Params Box	Only one One Shot device can be set at a time.
Any other combination of one or more devices or pins	General Delay Box	Any selected CLOCK, ONESHOT, SETUPHOLD, sub-circuit or other non-delay devices will be ignored for device delay calculations. Pin delays can be set on these devices.

NOTES:

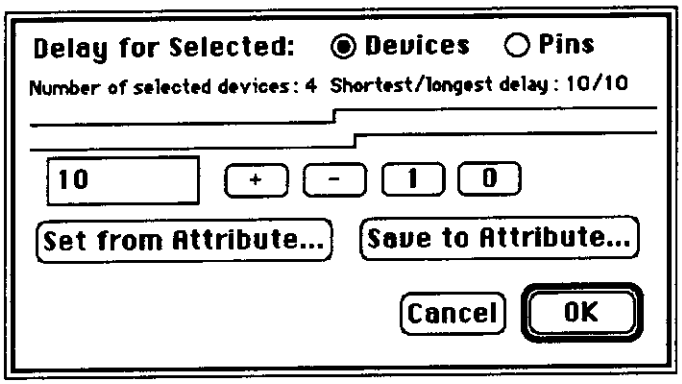
1) You cannot set the device delay of a sub-circuit device since its general delay characteristics are determined by its internal circuit. If any sub-circuit devices are selected, they will be ignored for device delay purposes. You can set the pin delay on sub-circuit devices to modify the path delay through a particular pin. Delays on the devices or pins inside a sub-circuit device are not affected by any settings on the parent device using this command.

2) The Simulation Params command relies on numeric information in a specific format being present in the Delay.Dev or Delay.Pin attribute fields. Any invalid information in these fields will be ignored and default values used.

See more information on the usage of specific attribute fields for simulation parameters in Appendix B - Simulation Attribute Fields

General Delay Box

For any collection of devices and pins with delay characteristics, the following box is displayed:



The controls in this box are summarized in the following table.

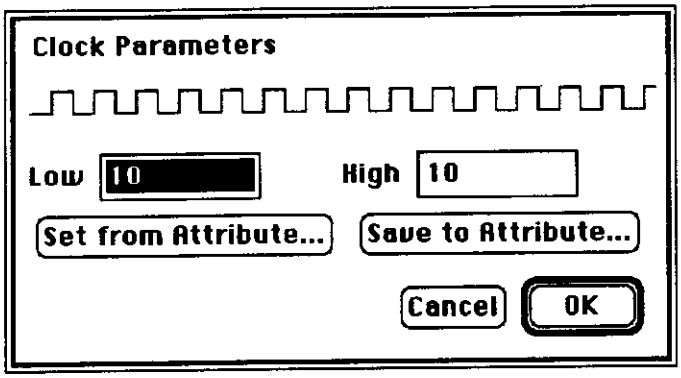
Devices	When this button is enabled, the other controls display and set the *device delay* characteristic of the devices currently selected in the circuit. Items such as CLOCK or sub-circuit devices which have no device delay characteristic will be skipped.
Pins	When this button is enabled, the other controls display and set the *pin delay* characteristic of the pins currently selected in the circuit.

Number of selected devices/pins	This shows a count of the devices or pins that will be affected by changes made in this box.
Shortest/Longest delay	This shows the shortest and longest delays found in any of the selected devices or pins. (Note that each device or pin has only a single integer delay value associated with it.)
Delay Text Box	If all selected devices or pins have the same delay value, it is shown in this box. If a variety of values exist among the selected items, this box will be empty. Typing a new value (between 0 and 32767) in this box will set all items to the given value.
+	Clicking this button will add 1 to the delays in all selected items, to a maximum value of 32767. .
-	Clicking this button will subtract 1 from the delays in all selected items, to a minimum value of zero.
1	Clicking this button will set the delay in all selected items to 1.
0	Clicking this button will set the delay in all selected items to zero.
Set From Attribute	Clicking this button will display the Set From Attribute box, allowing the delays in all selected objects to be loaded from a selected field in each object. This box is described below.
Save To Attribute	Clicking this button will display the Save To Attribute box, allowing the delays in all selected objects to be saved to a selected field in each object. This box is described below.

See Chapter III - Simulation for more information on the meaning and usage of device and pin delays.

Clock Params Box

When a single clock device is selected, the following parameters box is displayed:



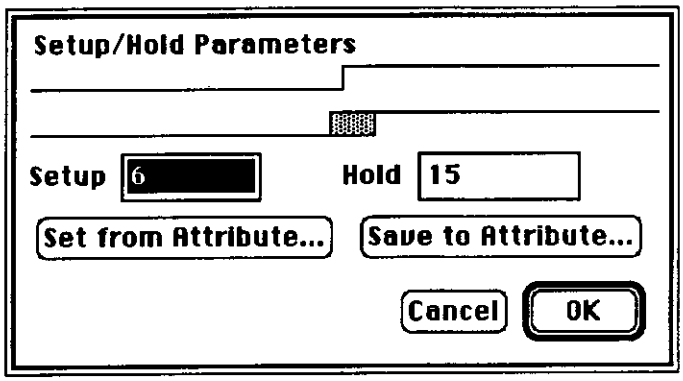
The controls in this box are summarized in the following table.

Low	This text box allows you to edit the low time setting of the selected clock device. Allowable settings are in the range 1 to 32767.
High	This text box allows you to edit the high time setting of the selected clock device. Allowable settings are in the range 1 to 32767.
Set From Attribute	Clicking this button will display the Set From Attribute box, allowing the parameters in the selected clock device to be loaded from a selected field. This box is described below.
Save To Attribute	Clicking this button will display the Save To Attribute box, allowing the clock parameter in the selected device to be saved to a selected field. This box is described below.

See Chapter VII - Primitive Devices for more information on how you can set the startup delay and initial value of a CLOCK device by setting the pin delay and inversion on the output pin.

SetupHold Params Box

When a single SETUPHOLD device is selected, the following parameters box is displayed:



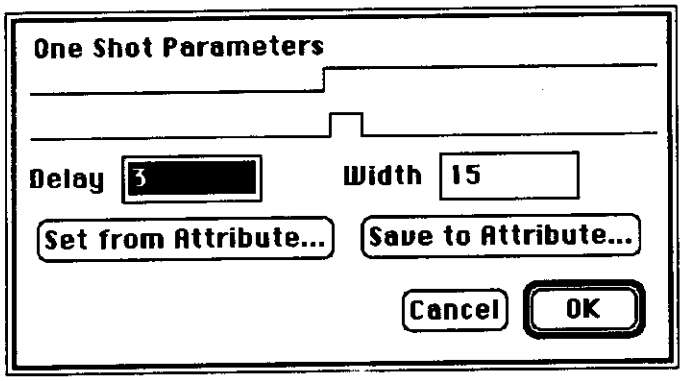
The controls in this box are summarized in the following table.

Setup	This text box allows you to edit the setup time setting of the selected device. Allowable settings are in the range 0 to 32767.
Hold	This text box allows you to edit the hold time setting of the selected clock device. Allowable settings are in the range 0 to 32767.
Set From Attribute	Clicking this button will display the Set From Attribute box, allowing the parameters in the selected device to be loaded from a selected field. This box is described below.
Save To Attribute	Clicking this button will display the Save To Attribute box, allowing the setup and hold parameters in the selected device to be saved to a selected field. This box is described below.

See Chapter VII - Primitive Devices for more information on how you can set the initial value of a SETUPHOLD device by setting the inversion on the output pin.

One Shot Params Box

When a single ONESHOT device is selected, the following parameters box is displayed:



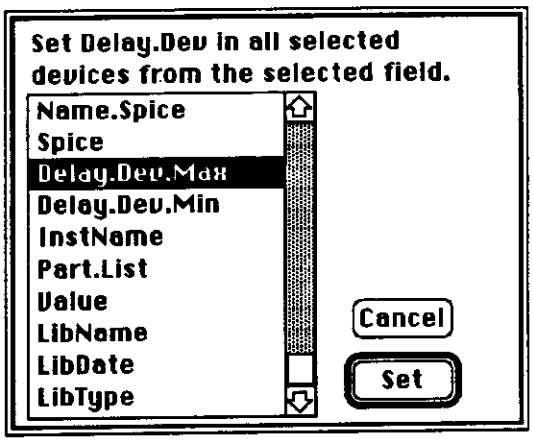
The controls in this box are summarized in the following table.

Delay	This text box allows you to edit the delay time setting of the selected device. Allowable settings are in the range 0 to 32767.
Width	This text box allows you to edit the width time setting of the selected clock device. Allowable settings are in the range 1 to 32767.
Set From Attribute	Clicking this button will display the Set From Attribute box, allowing the parameters in the selected device to be loaded from a selected field. This box is described below.
Save To Attribute	Clicking this button will display the Save To Attribute box, allowing the parameters in the selected device to be saved to a selected field. This box is described below.

See Chapter VII - Primitive Devices for more information on how you can set the initial value of a ONESHOT device by setting the inversion on the output pin.

Loading Delays from Attributes

Each of the simulation parameters boxes described above has Save to Attribute and Set From Attribute buttons. Selecting either of these buttons displays a box like the following (the Set button in this picture will appear as Save if the Save to Attribute option was clicked):



Set From Attribute

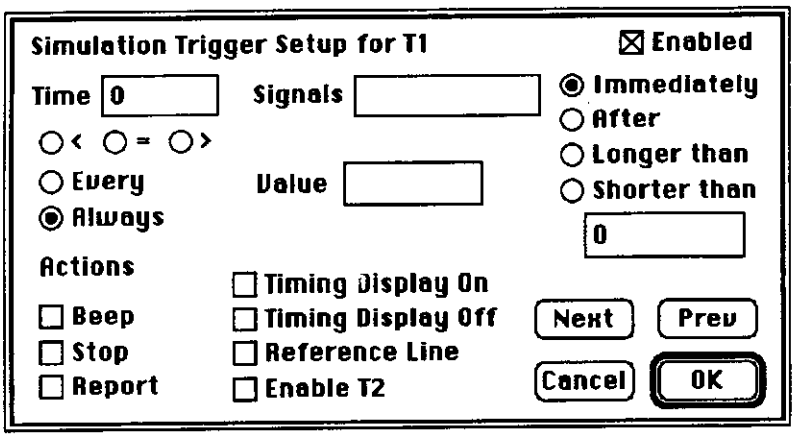
Selecting an attribute field in the list and clicking on the Set button will cause the contents of the selected field to be copied to the Delay.Dev or Delay.Pin field of all selected devices or pins. The copied value will completely replace the old contents. If the selected field is empty in a given object, the Delay.Dev or Delay.Pin field is left unchanged in that object.

Save to Attribute

Selecting an attribute field in the list and clicking on the Save button will cause the contents of the Delay.Dev or Delay.Pin field to be copied to the selected field in all selected devices or pins. The copied value will completely replace the old contents, even if the source value is empty.

Triggers

This command displays the trigger setup box, as follows:



NOTE: The trigger setup box can also be displayed by clicking on the Triggers button on the timing tool palette.

Trigger Conditions

A trigger is activated when three sets of conditions are met:

- The time condition, i.e. the current simulator time value is less than, equal to, greater than, or a multiple of, a given value.
- Signal value condition, i.e. one or more signals are at specified levels.
- The delay condition, i.e. the trigger is activated after a certain delay, or the signal condition exists for greater or less than a specified amount of time.

Trigger Enabling

When the "Enabled" switch is on, the trigger is "armed" and the selected actions will take place as soon as the trigger's conditions are met. If this switch is off, this trigger is disabled until enabled by the previous trigger or by this switch being clicked.

Time Condition Controls

The controls related to the time condition are summarized in the following table.

Time	In this box you enter the time value as a decimal integer. The meaning of this value is determined by the switches below it.
<, =, >	These buttons indicate that the trigger will be activated when the simulation time is less than, equal to, or greater than the given value, respectively.
Every	This time option specifies that the trigger will be activated every time the simulator time equals a multiple of the specified value.
Always	This specifies that the time condition should be considered to be always true. The time value is ignored.

Signal Condition Controls

The controls related to the signal condition are summarized in the following table.

Signals	In this text box, you can type the names of one or more signals whose values will be compared to the hexadecimal integer value typed in the Value box. One or more signals can be entered using the following formats: CLK — The single signal CLK D7..0 — The signals D7 (most significant bit), D6, D5 ... D0 IN1 OUT3 — The signals IN1 and OUT3
Value	In this box you enter the signal comparison value as a hexadecimal integer. This value is converted to binary and compared bit-for-bit with the signals named in the Signals box. The rightmost signal name is compared with the least significant bit of the value, etc.

Delay Condition Controls

The controls related to the time condition are summarized in the following table.

Immediately	This selection effectively disables the delay condition. The delay value entered in the box is ignored.
After	With this option selected, the trigger will be activated the specified amount of time after the time and signal value conditions are met, regardless of whether they continue to be true.
Longer Than	This delay option specifies that the trigger will be activated after the specified delay, as long as the time and signal conditions are still true.
Shorter Than	With this option selected, the trigger is activated if the time and signal conditions are true for less than the specified delay. I.e. Activation occurs when the time and signal conditions cease to be true if they have been true for less than the specified value.
Delay	The delay value is typed into this box as a decimal integer.

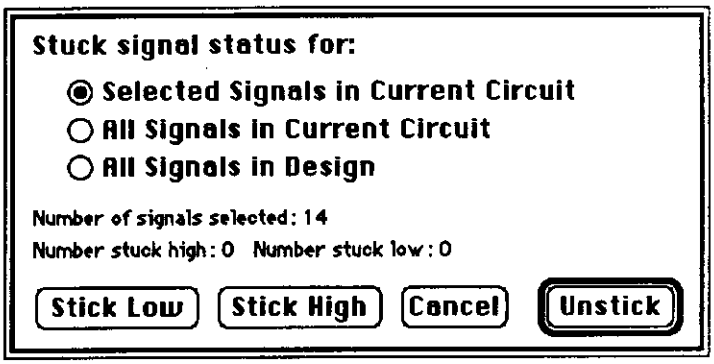
Trigger Actions

When a trigger is activated, any combination of the displayed actions can be invoked.

Beep	Generates a single system beep.
Stop	Stops the simulator immediately.
Display On	Turns on the timing waveform display.
Display Off	Turns off the timing waveform display.
Reference Line	Draws a reference line at this time on the timing waveform display.
Enable T_{N+1}	Enables the next numbered trigger.

Stick Signals

This command allows you to set the "stuck" status of the selected signals. It displays the following box:



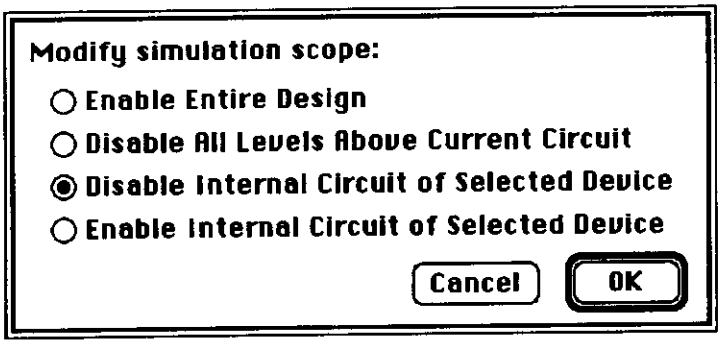
The controls in the box are summarized in the following table.

Selected Signals in Current Circuit	If this option is selected, the Stick High, Stick Low or Unstick option will apply only to signals currently selected in the schematic diagram.
All Signals in Current Circuit	If this option is selected, the Stick High, Stick Low or Unstick option will apply to all signals in the circuit represented in the topmost schematic window, including circuit pages in the same circuit. If this is part of a hierarchical design, only this circuit level is affected.
All Signals in Design	If this option is selected, the Stick High, Stick Low or Unstick option will apply to all signals in all parts of the current design.
Number of signals selected	This displays the number of signals that will be affected by any changes made in this box.
Number stuck high	This displays the number of signals in the selected scope that are currently stuck at a high level.
Number stuck low	This displays the number of signals in the selected scope that are currently stuck low.
Stick Low	This closes the box and applies a "stuck low" value to all selected signals.
Stick High	This closes the box and applies a "stuck high" value to all selected signals.
Unstick	This unsticks all selected signals, allowing them to return to their driven value.

See more information on stuck signal values in Chapter III - Simulation.

Simulation Scope

This command sets the scope of the simulation in a hierarchical design, i.e. to enable or disable parts of the design. The following box will be displayed:



The following table describes the actions available.

Enable Entire Design	Removes all previous disables and enables simulation of all parts of the design.
Disable All Levels Above Current Circuit	The current circuit (i.e. the one displayed in the frontmost circuit window) is set to be the topmost level of the hierarchy that will be simulated. Simulation is disabled in all other hierarchy levels and all driving port connectors at this level will drive at a high impedance level.
Disable Internal Circuit of Selected Device	The internal circuit of the device selected on the schematic will be disabled. All the output pins on the device will be set to a high impedance drive level.
Enable Internal Circuit of Selected Device	The internal circuit of the device selected on the schematic will be re-enabled. Its outputs will revert to the state determined by the internal circuit.

Clear Simulation

Selecting this item clears the timing window (if open), removes all scheduled signal events, sets all devices, signals and pins to their specified initial values (if any) and recalculates output values for all circuit elements.

Clear Unknowns

Selecting this item clears all flip-flop, counter and register primitives to the zero state, and attempts to remove all unknown signal values from the circuit. Note that certain circuit conditions may prevent signals from being placed in a known state:

- unconnected inputs that have not be set to a known level.
- storage devices, such as RAMs that have an unknown stored value.
- any simulation model that does not produce a known output when all inputs are known.

Signal Probe

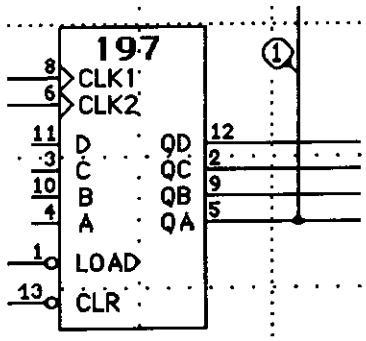
This command places the schematic in Signal Probe mode, allowing you to display and inject signal and pin values. When the probe tip is clicked and held on a signal line or pin, the cursor will show the current value on the signal or pin and track changes that occur as the simulation progresses.

NOTE: The Signal Probe tool can also be selected by the following means:

- Hold the Option key while selecting the Attribute Probe item in the Edit menu, OR clicking on the ? tool on the Schematic tool palette.
- Click the Probe button on the Timing tool palette.

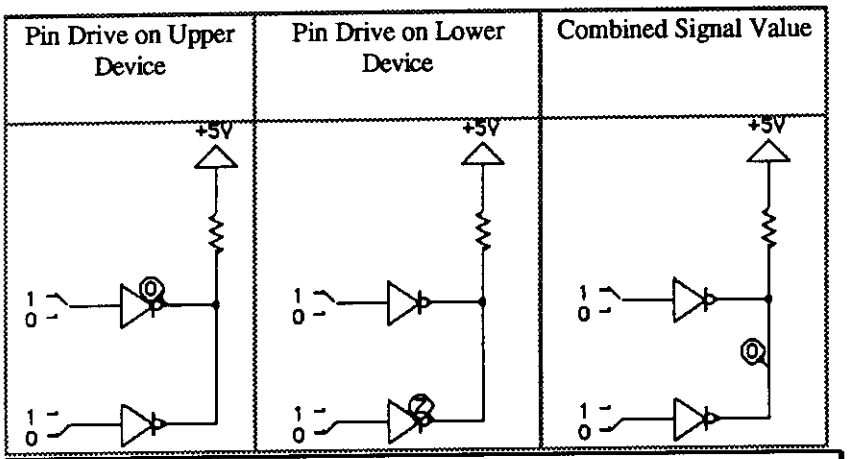
Probing a Signal

Only the signal under the cursor at the time of the click is examined; moving the mouse while the button is pressed does not change the signal being viewed.



Probing a Pin

If the probe tip is clicked on a device pin close to the device body, the probe shows the driving level of that pin, rather than the state of the attached signal. This can be used to resolve drive conflicts in multiple drive situations, as in the following example:



NOTE: The probe display does not distinguish between low and high drive levels.

Injecting a Value Using the Probe Tool

While the mouse button is held, you can press keys on the keyboard to inject new values onto a signal, as follows:

0	LOW.F
1	HIGH.F
X	DONT01.F
C	CONF.F
Z	HIGHZ
L	LOW.F stuck
H	HIGH.F stuck
space	Unstick

If a stuck value is forced onto a signal, it will not change state until the stuck value is cleared by some user action, regardless of device outputs driving the line. If a non-stuck value is forced, the signal value will revert to its appropriate new level when any change occurs on a device output driving the line.

The spacebar "unstick" command causes the signal to revert to its driven value.

See also the Stick Signals command in this chapter and Chapter III - Simulation for information about stuck values.

Timing Window Commands

Add to Timing

This command adds all selected signals in the current circuit to the timing display. If any selected items are unnamed or they are already displayed, they will be ignored. New items are added at the bottom of the timing display and will be selected after the add.

Add as Group

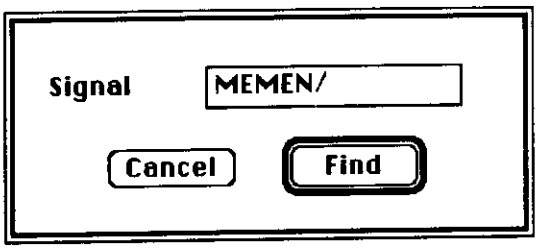
This command is similar to the Add to Timing command except that the selected items are all added as a single group. Where possible, items will be sorted in alphanumeric order with the lowest numbered item in the least significant bit position.

Auto Add

When this item is checked, any signals added or edited on the schematic will automatically be added to the timing window.

Find in Timing

This command allows you to locate a signal or group in the timing display by its label. The following box will be displayed:



The name of any displayed trace can be typed into the Name box. When the Find button is clicked, the item will be located and selected and the window will be scrolled vertically to display it. Note that the search is done by the displayed label in the timing window, not the original name of the item in the schematic.

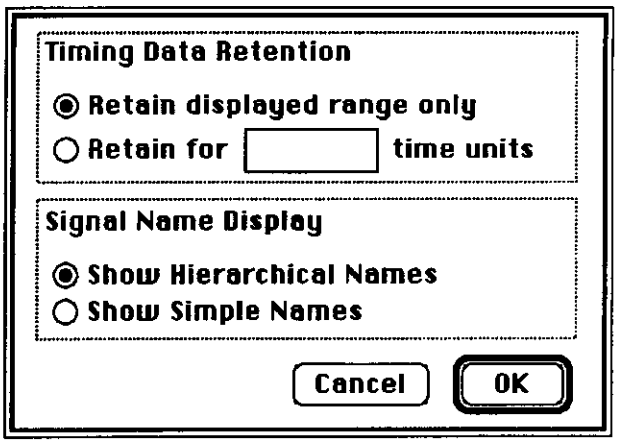
The following "wildcard" characters can be used in this search:

*	Matches any string of zero or more characters. For example "D*" will match "D", "D0", "D12", "DogGone", etc. "*" by itself matches anything.
#	Matches any string of 1 or more digits. E.g. "XA#" will match "XA0", "XA123", etc.
?	Matches any single character. For example, "CTRL?" will match "CTRLX", "CTRL/", etc.

The wildcard characters can be used in combination, as in "DATA#??" which will match "DATA123", "DATA1SQ", etc.

Timing Options

This command displays the following options box:



Timing Data Retention

These options allow you to determine how much signal event data is retained in memory when a simulation is run.

Each time a signal level change occurs DesignWorks creates a record in memory containing a reference to the signal, time, new value and source of the change. In a large simulation these records can consume enormous amounts of memory. This data can be retained for the following purposes:

- for use in refreshing the timing window should it become hidden and then be redisplayed.
- for use in timing window editing operations such as taking the output from one circuit and using it as stimulus for another.

Note that data can be retained only for signals displayed in the Timing window. Signal event data for all other signals is discarded immediately after it is no longer required for simulation.

The option "Retain for Displayed Range Only" is the normal default and results in data being discarded immediately after the corresponding point on the timing display scrolls off the left hand side. This results in minimal memory usage. The setting is equivalent to entering 0 in the retain time box.

The option "Retain for x Time Units" allows you to keep the signal event data for the specified amount of time after it scrolls off the left side of the

screen. If this results in a memory shortage occurring then the simulation will stop and a message will be displayed.

Signal Name Display

This option lets you choose how names of signals in sub-circuits will be displayed on the timing diagram. The option "Show Hierarchical Names" will display the complete "path" to the signal, i.e. prefixed with the names of all parent devices. The option "Show Simple Names" will show only the basic signal name in the label area of the diagram.

Show/Hide Timing Palette

This command displays or removes the timing tool palette. It can be invoked without the timing window displayed to make use of the current time display and simulation controls.

See Chapter V - The Timing Diagram for more information on the timing tool palette.

Timing Diagram Pop-Up Menu

Holding the command (⌘) key pressed while clicking in the label area of the timing diagram will display the timing pop-up menu.

These menu functions control the display of selected signals in the timing diagram and are summarized in the following sections.

Get Info

For groups, this command displays a box allowing reordering of the signals in the group. This affects the way the combined integer value is shown in the timing display.

For individual signals, a signal info box is displayed.

Go To Schematic

This command displays the signal corresponding to this trace in the schematic. If the trace is a group, it shows the corresponding bus, if any, or the first signal in the group.

Remove

This command removes the selected traces from the timing window. THIS CANNOT BE UNDONE!

Group

Combines all the selected traces into a single display group. If any of the selected traces were already grouped, they are in effect Ungrouped first and then recombined with other selected items into a single new group.

Ungroup

Breaks all signals in selected groups into individual traces.

Collect

Brings all selected items together in the display underneath the topmost selected item. This is used to bring associated signals closer together for easier comparison of timing, etc.

To Top

Sends all selected traces to the top of the timing diagram.

To Bottom

Sends all selected traces to the bottom of the timing diagram.

Appendix A - Timing Text Data Format

When a Copy or Cut operation is done on a selected area in the Timing window, two types of data are placed on the system clipboard:

- A picture of the selected area, in standard PICT format.
- A text description of the signal value changes occurring in the selected area.

This Appendix describes this text data format.

General Description of Format

The clipboard text format used by the Timing tool is actually a subset of the command structure supported by the TestPanel tool. Thus, it is always possible to paste timing data into the TestPanel. Copying data from the TestPanel tool to the timing diagram is also possible with some restrictions. The Timing tool does not support control commands like $REPEAT/$END which are available in the TestPanel.

NOTE: The TestPanel's Copy to Timing command is an alternate way of moving TestPanel data to the timing window. This bypasses the clipboard and does support control commands.

Format Specifics

The following rules describe the timing text data format:

- The data is pure ASCII text with no special binary codes except for standard tab and carriage return characters.
- The format of the data is based on the common "spreadsheet" text data format, i.e. each text item is followed by a tab character, except for the last one on a line which is followed by a carriage return.
- Every line has the same number of text items on it.
- The first line of the text (i.e. up to the first carriage return) is a "header" which indicates the meaning of the items on the following lines, by position.
- The lines following the header are signal value lines. Each line represents one time step. A complete data line is written out each time any value on the line changes. No line is written out for time steps in which none of the represented signals changed value.

Header Format

The header consists of a series of "commands", each starting with a $, which describe the meaning of the corresponding data items on the following lines.

The header always contains the command "$T" (denoting a time column), followed by a tab character, followed by "$D" (denoting a delay column). The remaining items depend on the traces that were selected in the timing diagram.

NOTE: The Timing tool always places the time and delay items in the order given here, although it will accept data with them in any order, or even completely missing. Since time and delay are redundant, either one is sufficient. If they are both missing a default delay value will be used.

Single Signal Items

An individual signal is specified by the text $I (for input) followed by a space, followed by the name of the signal. If the signal contains any blanks or control characters, it will be enclosed in quotation marks.

Grouped Items

Grouped items are denoted by the text "$I" followed by a blank, followed by the name of the group, followed immediately (without any spaces) by a list of the signals in the group, contained in square brackets. Any group or signal name which contains blanks or control characters will be enclosed in quotation marks.

NOTE: The TestPanel tool does not use group names, but will ignore any text immediately preceding the "[]" grouping brackets. For this reason, there must not be a space between the group name and the "[" or the group name will be interpreted by the TestPanel as a signal name.

Data Line Format

Each line following the header must contain one data item for each item in the header line. Thus, the first two items will always be:

- The time at which the events on this line take place. The Timing tool places the absolute time at which the events occurred (i.e. corresponding to the time scale on the diagram) in this column. However, when the data is pasted, the times are considered to be relative to the time of the first data line. This is a decimal integer which may take on any 32-bit unsigned value.

• The delay from this step to the next step. This is redundant information, since it can be derived from the times in the first column. It is provided for compatibility with TestPanel and for improved flexibility in exporting to outside software systems.

NOTES:

1) If the delay and time columns do not match, the longest time is used.

2) The delay on the last line has special significance because it indicates the delay from the last signal change to the end of the selected interval. When pasting, this value is used to determine how much time to insert.

The following items on a line will be signal or group values matching the items in the header.

• For groups or individual signals, the special values ".X" (for DONT01) and ".Z" (for HIGHZ) indicate that all signals covered by this data item are in the given state.

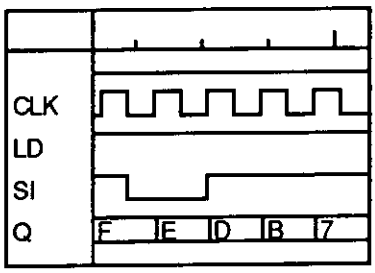
• Individual signals not in Don't Know or High Impedance states will be either "0" or "1".

• Grouped signals which are not all unknown or all high impedance will be specified by a hexadecimal value. The least significant bit of the value corresponds to the rightmost signal in the group list. Special character "X" may be substituted for a hex digit if any of the four signals represented by that digit is unknown, or "Z" if all the signals represented by that digit were high impedance.

Timing Text Example

Following is an example of the timing text data and the corresponding timing diagram.

$T	$D	$I Q[Q0 Q1 Q2 Q3]	$I SI	$I LD	$I CLK
87410	2	F	1	0	0
87412	10	F	1	0	1
87422	10	F	0	0	0
87432	1	F	0	0	1
87433	9	E	0	0	1
87442	10	E	0	0	0
87452	1	E	1	0	1
87453	9	D	1	0	1
87462	10	D	1	0	0
87472	1	D	1	0	1
87473	9	B	1	0	1
87482	10	B	1	0	0
87492	1	B	1	0	1
87493	9	7	1	0	1
87502	4	7	1	0	0



Appendix B - Simulation Attribute Fields

The tools making up the Simulator package make use of a number of attribute fields to store device, signal and pin parameters and simulation setup information. The following table summarizes the usage of these fields.

The following information is provided in the attribute table:

Field Name	The name of the field as it appears in the Define Attributes box.
Type	The type of objects this field is used in.
Tool	The name of the tool that uses this field. If this is blank, then the field is provided as a convenience for the user but not "hard-wired" to any program function.
Inst/Def	Specifies whether the field data is kept with the definition or instance of a sub-circuit. I.e. Instance fields can have a different value in each copy of a sub-circuit.
Description	How the field is used.

Field Name	Type	Tool	Inst/Def	Description
ABELSrcName	Device	FromABEL	Def	Contains the name of the MacABEL source file used to generate this part definition.
Delay.Dev	Device	Sim	Def	Specifies device delay. For most devices, a single decimal integer 0 to 32,767. For Clock, One Shot and SetupHold devices, two integers separated by commas. Should be set using the Simulation Params command and not edited manually.
Delay.Dev.Max	Device		Def	Maximum device delay. This is provided for use with the Load From Attribute button in the Simulation Params command. Not used internally.
Delay.Dev.Min	Device		Def	Minimum device delay. See Delay.Dev.Max.

Delay.Dev.Typ	Device		Def	Typical device delay. See Delay.Dev.Max.
Delay.Pin	Pin	Sim	Inst	A decimal integer specifying pin delay in the range 0 to 32767. Should be set using the Simulation Params command and not edited manually.
Delay.Pin.Max	Pin		Def	Maximum pin delay. See Delay.Dev.Max.
Delay.Pin.Min	Pin		Def	Minimum pin delay. See Delay.Dev.Max.
Delay.Pin.Typ	Pin		Def	Typical pin delay. See Delay.Dev.Max.
DWLSrcName	Device	FromABEL	Def	Contains the name of the ".dwl" file used to generate this part definition.
DWLSrcPath	Device	FromABEL	Def	Contains the "path name" of the ".dwl" file used to generate this part definition, i.e. names of all nested folders, separated by ":".
ExtCctDate	Device	SimLoad/ Schematic	Def	The "last modified" date of the external circuit file, stored as an integer in 1/60 sec since Jan. 1, 1904. Should not be edited manually. See ExtCctName.
ExtCctName	Device	SimLoad/ Schematic	Def	The file name of the external circuit file. Should normally be set using the SimLoad tool or Attach External Sub-circuit command.
ExtCctPath	Device	SimLoad/ Schematic	Def	The "path name" of the external circuit file, i.e. names of all nested folders, separated by ":". See ExtCctName.
Initial.Pin	Pin	Sim	Inst	The initial value for the pin on Clear Simulation. Output pins only. One character: 0, 1, X or Z.
Initial.Sig	Signal	Sim	Inst	The initial value for the signal on Clear Simulation. One character: 0, 1, X or Z.
Invert.Pin	Pin	Sim	Def	Any non-empty value indicates pin inversion.

Sim.InputMap	Design	Sim	Def	Used to specify alternate input mapping for primitive devices. Must be exactly five characters, each one 0, 1 or X. First character specifies value to substitute for low input value, second is high, third is high impedance, fourth is Don't Know, fifth is Conflict. E.g. "011XX" will substitute a high value for high impedance. Default is "01XXX".
Sim.TrigSave	Design	Sim	Def	Used by the Simulator to store the current trigger setup. Should not be edited manually.
TestVectors.Cct	Design		Def	Used to store sets of test vectors for the design. Not used internally.
TestVectors.Dev	Device	TestPanel	Def	Used to store test vectors for the device. Loaded and save by the Place and Test Device and Save Test to Device commands in TestPanel.
Timing.Save	Design	Timing	Def	Used to save the current status of the timing window. Should not be edited manually.

Appendix C - Device Pin Types

Every device pin has a characteristic known as its *pin type*. The pin type is set when the part entry in the library is created and cannot be changed for individual device pins on the schematic.

Refer to the DesignWorks/Schematic User's Manual for information on how to set the pin type while creating a device symbol.

What Pin Types are Used For

For many general schematic editing purposes, the pin type will be unimportant an can be ignored. However, pin type settings are important in the following cases:

- The pin type of each pin is used by the simulator to select what type of output values are generated by a pin. E.g. An open collector output will not generate a HIGH drive level.
- Pin type information is required in many netlist file formats for FPGA layout and digital simulation.
- Correct pin type settings allow the ErrorFind tool to check for fanout and multiple drive situations.
- Other future analysis tools may use this information for timing and loading analysis.

Pin Types Table

The following table lists the function of each of the pin types available in DesignWorks. The Output Value Mapping column specifies how output values specified by the model are mapped to actual pin drive values.

Pin Type	Description	Initial Value	Output Value Mapping
IN	Input - this is the default for pins created using the DevEditor tool. This setting is used for all pins on discretes except those with some digital function. No output value can be placed on an input pin.	HIGHZ	No output drive allowed.
OUT	Output - always enabled.	DONT01	None

3STATE	Output - can be disabled (i.e. high-Z). Note: The three-state capability only exists for specific primitive types that have a three-state enable pin. For other types this will behave like OUT.	DONT01	None
BIDIR	Bidirectional	DONT01	None
OC	Open collector output - i.e. pulls down but not up	DONT0Z	HIGH maps to HIGHZ.
BUS	Bus pin - This does not represent a physical signal but is a graphical representation of a group of internal pins, each having its own type. Bus pins cannot have values and are not supported on primitive device types.	None	None.
LOW	Output - always driving low	LOW	All values converted to LOW.
HIGH	Output - always driving high	HIGH	All values converted to HIGH.
LTCHIN	Input to a transparent latch - this is used for calculating cumulative setup and hold times.	HIGHZ	No output drive allowed.
LTCHOUT	Output from a transparent latch - this is used for calculating cumulative setup and hold times.	DONT01	Same as OUT.
CLKIN	Input to an edge-triggered latch - this is used for calculating cumulative setup and hold times.	HIGHZ	No output drive allowed.
CLKOUT	Output from an edge-triggered latch - this is used for calculating cumulative setup and hold times.	DONT01	Same as OUT.
CLOCK	Clock input - this is used for calculating cumulative setup and hold times.	HIGHZ	No output drive allowed.
OE	Open emitter output - i.e. can pull up but not down.	DONT1Z	LOW maps to HIGHZ.
NC	A no-connect pin	HIGHZ	No output drive allowed.

Device Pin Type and Simulator Efficiency

Incorrect device pin type settings can have a major impact on simulation speed, even in cases where they wouldn't affect the correctness of the results.

Bidirectional Pins

The use of bidirectional pins should be avoided unless specifically required by circuit logic. On primitive types, any value change on a signal attached to a bidirectional pin will cause the device model to be called to reevaluate the device. On sub-circuit devices, the simulator must make several passes through all circuit levels which may affect the value of the signal or be affected by it. Setting a pin on a sub-circuit device to be an input or output greatly reduces this overhead.

Output Pins

If a device pin will only ever be used to drive the attached signal and the device cannot be affected by changes in value on the pin, then it should be an output type. Changes in the value of a signal attached to an output pin do not cause the device model to be called for reevaluation. This is particularly significant for sub-circuit devices.

Input Pins

Pins with an input type setting can never place a drive value on the attached signal. On sub-circuit devices this provides an important hint to the simulator that internal value changes on the attached signal will not affect any other circuit level.

Appendix D - Primitive Device Pin Summary

In DesignWorks primitive device types, the function of each pin is determined by its type (i.e. input or output) and its sequential position in the device's pin list (i.e. as seen when the part is opened in the DevEditor). Pin name is not significant. Each type has specific rules about the ordering of pins. Failure to adhere to these rules will result in incorrect simulator operation.

NOTE: Bus pins are not supported on primitive device types.

For many primitive types, certain control inputs and outputs can be omitted to create simplified device types. E.g. On flip-flop types, the Set and Reset inputs can be omitted. The table below shows which combinations of inputs are allowable and the required order.

Pin Inversion

In addition to the pin function options shown in the table, any pin on any device can be inverted by specifying a value in the Invert.Pin attribute field. Any non-empty value will cause the pin logic to be inverted.

Pin Function Table

The following table lists the pin functions and orders for all primitive device types. In some cases, a number of pins can be optionally omitted, so rules are given rather than enumerating all possible combinations.

See Chapter VII - Primitive Devices for more information.

Primitive Type	Limitations	Pin Names and Types	Possible Pin Orders
NOT	Exactly 1 input and 1 output	IN - in OUT - out	1) IN OUT
AND, NAND, OR, NOR, XOR, XNOR	N inputs: $1 \leq N < 799$	$IN_0..IN_{N-1}$ - in OUT - out	1) $IN_0..IN_{N-1}$ OUT

X-Gate	Exactly 2 ports and 1 enable	X1 X2 - bidir EN - in	1) X1 EN X2
Buffer	N data inputs N data outputs $1 \le N < 256$	$IN_0..IN_{N-1}$ - in $OUT_0..OUT_{N-1}$ - out EN - in	1) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ EN 2) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$
Resistor	Exactly 2 pins	X1 X2 - bidir	1) X1 X2
Multiplexer	L select inputs M output bits $1 \le M \le 256$ N inputs/ output $2^{L-1} < N \le 2^L$ i.e. the number of inputs per output bit can be less than the number of select input combinations	$S_0..S_{L-1}$ - in $IN_{0,0}..IN_{N-1,M-1}$ - in * EN - in § $OUT_0..OUT_{M-1}$ - out * $IN_{n,m}$ is the input routed to output m when select value is n. § An enable input can exist only if $N == 2^L$, otherwise the extra input is assumed to be a data input.	1) $IN_{0,0}..IN_{0,M-1}$ $IN_{1,0}..IN_{1,M-1}$ .. $IN_{N-1,0}..IN_{N-1,M-1}$ $S_0..S_{L-1}$ $OUT_0..OUT_{M-1}$ 2) $IN_{0,0}..IN_{0,M-1}$ $IN_{1,0}..IN_{1,M-1}$ .. $IN_{0,0}..IN_{N-1,M-1}$ $S_0..S_{L-1}$ EN $OUT_0..OUT_{M-1}$ § § Option 2 only if $N == 2^L$
Decoder	L select inputs M output bits $1 \le M \le 256$ AND $2^{L-1} < M \le 2^L$ i.e. the number of output bits can be less than the number of select input combinations	$S_0..S_{L-1}$ - in EN - in $OUT_0..OUT_{M-1}$ - out	1) $OUT_0..OUT_{M-1}$ $S_0..S_{L-1}$ 2) $OUT_0..OUT_{M-1}$ $S_0..S_{L-1}$ EN
Adder, Subtracter	N output bits N "A" operand inputs required N "B" operand inputs optional $1 \le N \le 256$	$A_0..A_{N-1}$ - in $B_0..B_{N-1}$ - in CIN - in $SUM_0..SUM_{N-1}$ - out COUT - out	1) $A_0..A_{N-1}$ $B_0..B_{N-1}$ $SUM_0..SUM_{N-1}$ CIN COUT * * $B_0..B_{N-1}$ CIN & COUT can be omitted in any combination

D Flip-Flop, D Latch	Must have at least D and CLK inputs and Q output	S - set in D - D in C - clock in R - reset in Q - out NQ - inverted out	1) S D C R Q NQ 2) S D C R Q 3) D C R Q NQ 4) D C R Q 5) D C Q NQ 6) D C Q
JK flip-flop	Must have at least CLK input and Q output	S - set in J - J in K - K in C - clock in R - reset in Q - out NQ - inverted out	1) S J C K R Q NQ § 2) S T C R Q NQ *§ 3) T C R Q NQ *§ 4) C R Q NQ § 5) C Q NQ § § NQ can always be omitted * T = J & K tied together
Register	N output bits N input bits $1 \le N \le 256$	$IN_0..IN_{N-1}$ - in CLK - in CLR - in $OUT_0..OUT_{N-1}$ - out	1) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ CLK CLR 2) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ CLK
Counter	N output bits N input bits (optional) $1 \le N \le 256$	$IN_0..IN_{N-1}$ - in CLK - in LD - in CLR - in UP - in EN - in $OUT_0..OUT_{N-1}$ - out COUT - out	1) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ CLK LD CLR UP EN COUT 2) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ CLK LD CLR UP COUT 3) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ CLK LD CLR COUT 4) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ CLK LD COUT 5) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ CLK COUT Note: $IN_0..IN_{N-1}$ & COUT can always be omitted
Shift Register	N output bits N input bits $1 \le N \le 256$	$IN_0..IN_{N-1}$ - in CLK - in LD - in CIN - in $OUT_0..OUT_{N-1}$ - out	1) $IN_0..IN_{N-1}$ $OUT_0..OUT_{N-1}$ CLK LD CIN 2) $OUT_0..OUT_{N-1}$ CLK CIN
One Shot		CLK - in CLR - in Q - out NQ - out	1) CLK CLR Q NQ 2) CLK CLR Q

Clock Osc	Exactly one output pin	CLK - bidir	1) CLK
Binary Switch	Exactly one pin	SW - bidir	1) SW
SPST Switch	Exactly 2 pins	X1 X2 - bidir	1) X1 X2
SPDT Switch	Exactly 3 pins	X1 X2 COM - bidir	1) X1 X2 COM
Probe	Exactly 1 pin	PR	1) PR
Hex Keyboard	4 or 5 pins	$X_0..X_3$ - bidir STROBE - out	1) $X_0..X_3$ STROBE 2) $X_0..X_3$
Hex Display	Exactly 4 pins	$X_0..X_3$ - in	1) $X_0..X_3$
SetupHold	Exactly 3 pins	CLK - in DATA - in Q - out	1) CLK DATA Q
Unknown Detector	Exactly 2 pins	D - in Q - out	1) D Q

Credits

DesignWorks, LogicWorks and MEDA are trademarks of Capilano Computing Systems Ltd.

The DesignWorks Simulator were designed by Chris Dewhurst and coded by Chris Dewhurst and Addam Smith of Capilano Computing Systems Ltd.

This manual was written by Chris Dewhurst using Microsoft Word (a trademark of Microsoft Corporation of Bellevue, Washington).

Macintosh is a trademark of Apple Computer, Inc. of Cupertino, California.

MacABEL is a trademark of Data I/O Corporation of Redmond, Washington.

Any problems or questions can be referred to:

Technical Support,
Capilano Computing Systems Ltd.,
406 - 960 Quayside Drive,
New Westminster, B. C.
Canada, V3M 6G2
(604) 522-6200
Fax (604) 522-3972

or on AppleLink:

D0816

This page is intentionally left blank.

Index

Page number in bold print indicates primary reference to that topic.

$DELAY command 50, **55**, **56**, 64, 134
  example 71, 136
$EXPECTED command 50, **53**, 63, 64
  example 70
$INPUTS command 50, **51**, 134
  example 56, 70, 72, 136
$OUTPUTS command 50
  example 56
$REPEAT command **56**, 58, 66, 133
  example 72
$STOP command 56
$TIME command 50, **54**, 65, 134
$WAIT command 50, **54**, 58
  example 71
++ operator 52
  example 56
-- operator 53
.C value code 51, **52**, 53, 63
.DC value code 54
  example 71
.dwl file **101**, 104, 105, 106, 108
.N value code **52**, 53, 63, 70
  example 70
.X value code
  TestPanel **52**, 53
  Timing text data 135
.Z value code
  TestPanel 52, 53
  Timing text data 135
<> zoom option 37
>< zoom option 37
ABELSrcName attribute field 137
Add As Group command 35, **128**
Add To Timing command 11, 34, 35, **128**
adder device **82**, 144
AND device **75**, 143
attributes
  delay 14, 119
  loading CLOCK parameters from 117
  loading delay from 116
  loading ONESHOT parameters from 119
  loading SETUPHOLD parameters from 118
  loading test vectors from 58, 67
  pin inversion 23, 76, 143
  PLDs 106
  PROM 103
  RAM 96, 97, 98
  saving CLOCK parameters to 117
  saving delay to 116
  saving ONESHOT parameters to 119
  saving SETUPHOLD parameters to 118
  saving test vectors to 57, 68
  setting in DevEditor 76
  table 137
  value initialization 8, 18, 19
Auto Add command **129**
bidirectional pins 141
  efficiency 23, 142
  initial values 20
  on sub-circuit devices 22, 23
  PLDs 106
  transmission gate 77
binary switch device **91**, 146
Boolean formulas 99
buffer device 13, **77**, 144
  non-inverting 78
  PLDs 99, 100
buffering sub-circuit pins 15, 22
bus pins 12
busses 11
  bus pins 12, 141
  open collector 9
  Timing 132
capacitance 6, 11, 78
Clear command
  TestPanel 60
  Timing 37, 40, 112
Clear Sheet command 66

Clear Simulation command 18, 37, 84, 89, 109, **126**, 138
Clear Unknowns command 8, 19, 37, 84, **126**
Clear X option 37
clipboard
  copying Timing to TestPanel 58
  TestPanel data format 59
  TestPanel edit commands 48, 59
  Timing edit commands 40, 110
  Timing picture 41
  Timing text format 133
clock device 5, 12, 18, 46, **89**, 146
  delay attribute format 14
  setting parameters 90, 114, 117
  synchronizing 89
clock pulse
  TestPanel 51, 52, 53, 63, 70
clocked devices 14, 15, 70, 83, 85, 93
Collect command 34, 132
conflict logic state 7, **9**, 11, 15, 19, 42, 75, 92, 93, 127
Copy command
  TestPanel 59
  Timing 37, 40, 111
Copy To Timing command 44, 45, 64, **66**, 133
counter device 14, 19, **87**, 126, 145
Cut command
  TestPanel 59
  Timing 37, 40, 110
D flip-flop device 14, 19, **71**, **83**, 145
D latch device **84**, 145
decoder device **81**, 144
decrementer device 83
Default Header command **64**, 69
delay **12**
  attributes 14
  clock pin 90
  devices 115
  effect of zero delay 13
  pins **12**, 115
  PLDs 106
  primitive devices 12
  setting 13, 115
  sub-circuit devices 12
  time units 6
Delay.Dev attribute field **14**, 115, 120, 137
Delay.Dev.Max attribute field 14, 137
Delay.Dev.Min attribute field 14, 137
Delay.Dev.Typ attribute field 14, 138
Delay.Pin attribute field **14**, 23, 24, 96, 115, 120, 138
Delay.Pin.Max attribute field 14, 138
Delay.Pin.Min attribute field 14, 138
Delay.Pin.Typ attribute field 14, 138
Delete Cols command 61
Delete Rows command 61
Delete Time command 37, 41, 111, **113**
DesignWorks Link file 101
DevEditor
  attributes 76
DevEditor tool
  attributes 24
  inverted pins 17, 76, 77
  pin order 73, 76, 143
  pin type 22, 140
  PLDs 107
  PROM devices 103
  RAM devices 98
devices
  delay **12**
    setting 114
  effect of unknown inputs 75
  gates 75
  high impedance inputs 15
  initial value 20
  input value mapping 15
  input values 15
  pin delay 12
  pin inversion 17
  pin type 16, **140**
  primitive types 12, 25, **73**, 143
  setup and hold times 15, 53, 70, 83, 93, 118, 141
  simulation models 25
  simulation pseudo-devices 26
  sub-circuit devices 22
discrete components
  pin type 140
Disp On/Off option 18, 37, 123

Don't Know logic state **8**, 15, 19, 42, 75, 92, 135
  input value mapping 139
  RAM 96
  TestPanel 52, 53
Draw Sig command 112
Duplicate command 37, 41, **112**
DWLSrcName attribute field 138
DWLSrcPath attribute field 138
events **5**, 6, 13, 114
  clearing 18, 37, 126
  editing 40, 66, 110, 112
  retaining 130
  saving to file 57, 109, **110**
  settling 46, 62
  TestPanel 44
  text format 134
ExtCctDate attribute field 30, 138
ExtCctName attribute field 28, 138
ExtCctPath attribute field 138
external references 27
fall time 5
fanout 140
feedback 5, 13, 37, 99, 100, 106
file formats
  PLD (dwl) files 101
  timing text format 133
Fill Down command **60**, 70
Fill Right command 60
Find in Timing command **129**
flip-flops
  D-type 83
  Initializing 84
  JK-type 85
  setup and hold times 83
forcing logic states 6
FromABEL tool 3, 99, **104**
  errors 108
  unsupported devices 107
gates **75**, 143
Get Info command 35, **131**
Go To Schematic command 132
ground and power connections 15, 24
  PLDs 106
Group command 132
groups **35**
  adding to display 128
  bit order 35
    changing 35
  busses 35
  creating 35, 132
  drawing tools 41
  find by name 129
  group name 35
  ungroup 35, 132
header
  Default Header command 64
  TestPanel 44, 48, 52
  Timing text data 133
hex display device **93**, 146
hex keyboard device **92**, 146
Hide Timing Palette command 36, **131**
hierarchy **21**
  attributes 137
  delay 12
  editing an open circuit 21
  hierarchy mode 21
  Link to Circuit command 61
  pin delays 23
  pin type 22
  port connector pin type 23
  port interface 22
  signal names 131
  simulation scope 125
  stuck signals 124
  TestPanel 45
  timing window 33
high impedance 7, 15, 24, 75, 92, 125
  drawing on timing diagram 42
  in switches 78, 91
  in Timing text data 135
  input value mapping 15, 139
  TestPanel 52, 53
hold time (See setup and hold times)
incrementer device 82
Initial.Pin attribute field 18, 19, **20**, 84, 138
Initial.Sig attribute field 8, 18, 19, **20**
Insert Cols command 61
Insert Rows command 60
Insert Time command 37, 41, 113
installation 3

inversion **17**, 76, 143
Invert.Pin attribute field **17**, 23, 90, 96, 138, 143
Invert.Pin field 76
JK flip-flop device 14, 15, 19, **85**, 145
Karnaugh maps 100
Link to Circuit command 45, **61**
Load From Design Attr command 58
logic probe (See probe)
MacABEL 100, **104**
MEDA 5, 25, 107, 147
memory device See RAM or PROM
memory usage 6, 29, 73, 77, 130
  PROM 100
  RAM 95, 97
menus 109
  TestPanel 57
  Timing pop-up 131
models
  attaching 26
  creating 25, 31
  installation 4
  MEDA 25
  purging 30
Monitor Mode command 45, 57, **66**
multiple drive 10
multiplexer device **79**, 144
names
  external model file names 30
  port connector 32
  special names 0 and 1 15
NAND device **75**, 143
nanoseconds 6
NOR device **75**, 143
NOT device **75**, 143
One Shot device **91**, 114, 145
  setting 91, 119
open collector pins 7, 9, 10, 23, 78, 98, 103, 140, 141
open emitter pins 10, 23, 78, 141
Open Test File command **57**, 69
Open Timing Text command 109
OR device **75**, 143
oscillator 89
Page Setup command 110
Paste command
  TestPanel 58, **59**
  Timing 37, 40, 109, 111
pins
  bidirectional 142
  bus pins 141
  delay 12, 14, 22, 23, 83, 90
    on clock devices 89
    setting 114
  efficiency 141
  function table 143
  hierarchical block 23
  high impedance inputs 15
  initial values 19, **20**
  input 140, 142
  input value mapping 15
  inversion **17**, 76, 143
  order in primitive devices 143
  output 140, 142
  output value mapping 140
  pin type 16, **140**
PLA device **99**, 100
  size limits 100
Place and Test Device command **67**, 68, 139
PLCC package type 106
Point command 38, 112
port connector 12, 22, 24, **31**, 65, 125
  pin type 23
primitive devices 11, 25, **73**
  delay 12
  input value mapping 15, 139
  pin order 143
  test vectors 68
Primitive Gates library 75
Primitive I/O library 26
Print Timing command 110
probe
  cursor 11, **126**
  device **92**, 146
Probe option 37
Programmable Logic Devices **99**
PROM device **99**
  editing 103
  size limits 100
  viewing contents 104
PromPLA tool 3, 101, **102**

pulldown resistor 78
pullup resistor 11, 15, **78**
Quine-McClusky method 100
RAM device 15, **95**, 126
  editing 98
  size limits 97
RAM tool 3, **96**
register device 14, 15, 19, **85**, 145
Remove command 34, 132
Report tool 26, 28
Reset option 48
resistive logic states 7, **11**, 78
resistor device 11, 15, **78**, 144
Resize Columns command 67
resolution 33
Restart option 37
retention options 130
ring oscillator 8
rise time 5
Run command 113
Save Test File As command 70
Save Test File command **57**
Save Test to Device command **68**, 139
Save Timing Text As command 110
Save to Attribute option 14, 116, 117, 118, 119, 120
Save to Design Attr command 57
Select All command
  TestPanel 60
  Timing 39, **113**
Set from Attribute option 14, 116, 117, 118, 119, 120
Set Trace File command 57
setup and hold times **15**, 53, 70, 93, 118, 141
  flip-flops 83
SetupHold device 16, 83, **93**, 114, 137, 146
  setting 118
Shift key 26, 34, 38, 47, 58, 59, 92
shift register device 14, 15, **89**, 145
Show Timing Palette command 36, **131**
Signal Probe command **126**
signal probe tool 10, 11, 24, 37, **126**
signals
  0 and 1 24
  displaying in timing diag. 6, 128
  hierarchical names 131
  in triggers 122
  initial values 19, **20**
  logic states 6
  probing 126
  stuck value **9**, 124
Sim.InputMap attribute field 15, 139
Sim.TrigSave attribute field 139
SimLoad tool 3, 26, **27**
simulation **5**
  description 5
  speed 36, 113
Simulation menu 113
Simulation Params command 12, 13, **114**
  clock device 89, 90
  one shot 91
  One Shots 91
  SetupHold device 93
Simulation Scope command 125
simulation speed 36, 113
simulation time 13, 33, 36, 40, 41, 50
  triggers 122
Single Step command 37, **114**
SPDT switch device 15, **92**, 146
speed
  menu 113
  tool palette 36
Speed menu 113
SPST switch device 15, **91**, 146
Step option 37
Stick Signals command 10, **124**
Stop command 37, **113**
stuck signal values **9**
  0 and 1 signals 24
  clearing **10**, 124
  power and ground 24
  setting **10**, 63, 124
  signal probe 128
  Stick Signals command 124
  storage devices 14
  TestPanel 63
sub-circuits
  attributes 137
  checking external references 31
  creating external models 31

delay 12, 114, 115
external 25, **26**, 27, 138
input value mapping 15
internal 25
pin inversion 17
pin type 142
PLDs 99, 100
purging 30
reloading 30
signal names 131
SimLoad 27
simulation 25
TestPanel 45, 61, 65
updating 30
subtracter device **83**, 144
switch device
binary **91**, 146
SPDT 15, **92**, 146
SPST 15, **91**, 146
synchronizing
clocks 89
TestPanel 71
synchronous counter See counter device
Test Options command 52, 55, **62**, 71
test vectors **43**, 49
copying to timing diagram **66**
design 139
device 67, 139
example 69
TestPanel tool 3, **43**
clipboard data format 133
closing 47
commands 49
editing 47
examples 68
hierarchical designs 45
menus 57
multiple TestPanels 46
opening 46
TestVectors.Cct attribute field 139
TestVectors.Dev attribute field 68, 139
three-state buffer device **77**, 144
three-state outputs 7, 23, 141
PLDs 99
RAM 95
time
retention time 130
time units 6
tool palette display 36
time scale 33
Timing Data Retention option 130
timing diagram **33**
adding traces 34
clipboard data format 133
drawing tools 41
editing **37**, 41
reference lines 18, 123
removing traces 34, 132
resolution 33
status box 34
time scale 33
tool palette 34, 36, 131
Timing Options command **130**
timing text data format 133
Timing tool 3, **33**
Timing.Save attribute field 139
To Bottom command 34, 132
To Top command 34, 132
tool palette 131
controls 36
time display 36
timing 36
trademarks 147
transmission gate device **77**, 144
triggers 18, **121**
display on/off 37
setting **121**
setup and hold checking 83
Sim.TrigSave attribute field 139
Triggers command 37, **121**
Triggers option 37
unconnected inputs 7, 9, 19, 92, 126
Undo command 58, 110
Ungroup command 35, 41, **132**
unknown detector device 94, 146
Update Display command 67
X-Gate device **77**, 144
XNOR device **75**, 143
XOR device **75**, 143
zero delay **13**, 92, 106, 116
zoom in/out 37

DesignWorks™ 3.1 Update Notes

These notes provide supplementary information on DesignWorks version 3.1, released March 9, 1993 for users of DesignWorks™ 3.0.

NOTE: See the section later in these notes for information on the changes in the 3.1 version that are not covered by the DesignWorks/Schematic Version 3.0 Reference Manual included with this release.

What you have received

This package contains the materials necessary to upgrade owners of DesignWorks™ 3.0 to the 3.1 release, and should contain the following:

1 Update Notes

5 Program Diskettes

If you have the complete 8 disk set, then you should refer to the DesignWorks™ 3.1 Release and Installation Notes for instructions.

Update Installation Notes

WARNING: If you already have a DesignWorks 3.1 installed on your disk the Installer will completely remove and replace any folders in "DesignWorks™ 3.1" having the same names as folders in the new set. Move any files you want to keep to another location before installing the new software.

©1993 Capilano Computing Systems Ltd. All rights reserved.

Compatibility With DesignWorks™ 3.0

DesignWorks 3.0 and 3.1 are fully file compatible.

Device Symbol Library Selection

The DesignWorks 3.1 update includes the following device symbol libraries:

- A new set of "compact" symbols for reduced schematic size and a better match with the DesignWorks 2.x format.
- Updated versions of some of the libraries included with DesignWorks 3.0.
- A FixLib tool which updates all remaining libraries in the 3.0 release. An error existed in the packaging information for some of the library parts provided with the 3.0 release. This affected only parts which have multiple different symbols applicable to the same package. If you are continuing to use libraries supplied with 3.0 and this problem is a concern for you, please see the instructions below and also the notes on the FixLib tool in the Utilities folder. All libraries provided on the 3.1 update diskettes have been repaired.

You may decide to completely replace the 3.0 libraries with the new compact symbols, or you can update your existing libraries by following the instructions below.

Program Installation Procedure

Before starting the installation procedure you should take the following preparatory steps:

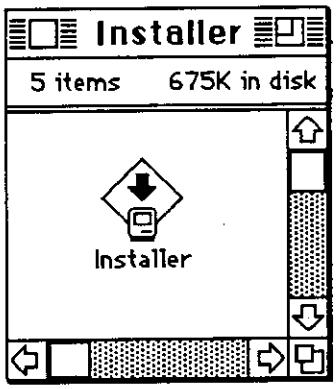
1) The Installer runs much more quickly and with less disk swaps if you disable all extensions on your machine before proceeding. With Macintosh System Software version 7.x, you can do this by selecting the Restart menu command in the Finder's Special menu and hold the Shift key pressed until you see the message "Extensions Off" in the startup screen.

2) The standard installation of DesignWorks and its default libraries requires about 19 megabytes of disk space. Most of this space is taken up by parts libraries. If you want to install the default system, you will need to ensure that this amount of space is available. You can use the "Customize" option to choose only a basic program installation to get started, or to install additional libraries. In any case, a minimal installation requires at least 5 megabytes.

NOTE: The DesignWorks release includes 3 complete sets of libraries in different symbol formats. By default, only the DesignWorks (compact) symbols are installed. You can alternatively install the ANSI or IEEE symbol libraries. You cannot install the ANSI and Compact libraries in a single operation because of file name conflicts. If you want to install both, you will have to install one set, then move or rename the folder, then install the second set.

To start the installation:

1) Insert the diskette labeled "Installer". You should see the following files on the diskette:



2) Do not copy any of these files onto your hard disk! Double-click on the file "Installer". This will start up the installer program.

3) At this stage there are a number of options:

• If you intend to use only the new "Compact" format symbols and you do not intend to use the standard ANSI or IEEE format symbols (large symbols):

Select the Easy Install option in the installer dialog. You may use the custom install option to get the SearchLib utility. This will create a complete DesignWorks™ 3.1 installation on your disk. This installation requires about 19 megabytes. You can reduce the size of the installation by using the Custom option to select only the required libraries. For example, you may elect to omit the 54XX libraries.

• To install the update without installing any new libraries:

You need to use the Custom install option in the installer dialog. This will give you a choice of packages to install. Select the DesignWorks package and, optionally, the SearchLib package. Do NOT install the ANSI or IEEE library packages that you see here at this point.

• To use the libraries from your existing DesignWorks 3.0 installation:

1) From your old Libs folder in the DesignWorks 3.0 folder open the ANSI and IEEE folders and drag the following folders into the trash:

75 TTL
CMOS
ECL
Memories
MicroProcessors
Miscellaneous
PLD

These libraries have replacements on the Update disks.

2) Drag the Libs folder from your DesignWorks 3.0 folder into your new DesignWorks 3.1 folder. If you have deleted the old DesignWorks 3.0 folder then you may install the libraries from your original DesignWorks 3.0 release disks by using the custom install. If you re-install the libraries you need only install the 54 and 74 TTL packages for the ANSI or IEEE symbols as needed.

3) Inside the DesignWorks™ 3.1 folder, move the FixLib module from the Utilities folder into the Tools folder.

4) Move the FixLibData folder in the Utilities folder into the DesignWorks™ 3.1 folder.

5) Launch DesignWorks 3.1 and select the FixLib tool from the Tools menu. This will step through all the installed files in your Libs folder and correct a problem that existed in the Units.All attribute in some of the parts.

6) Go back to the installer and use the Custom Install option to install the ANSI and/or IEEE Large format symbols found there. These packages replace the 75 TTL, CMOS, ECL, Memories, Micro Processors, Miscellaneous and PLD libraries.

Serial Number Entry

If you had an existing DesignWorks™ 3.x installation on your system, you should not have to re-enter the serial number. The same serial number provided with your earlier release still applies to the new version.

Correcting or Re-entering the Serial Number

If for any reason you should wish to re-enter the serial number information, you can hold the Option key pressed while starting the program from the Finder. The program will then redisplay the serial number entry box.

Troubleshooting Installation

PROBLEM: Program seems to run, but no windows appear and the Window and Tools menus have nothing in them, or tools are missing.

CAUSE: The program has failed to locate any of the tool files. This could be due to any of the following:

- The tools are not in the same folder as the DesignWorks™ 3.1 application or the Tools folder has been renamed.
- The DesignWorks Setup file has been renamed or is not in the same folder as the DesignWorks™ 3.1 application.
- The folder name specified in the TOOLFOLDER line of the DesignWorks Setup file is incorrect.

PROBLEM: Error message about "Could not open library XXXX"

CAUSE: The name or location of a library specified in the DesignWorks Setup file is incorrect.

PROBLEM: Error message about "Unexpected changes have been made to the application..." appears on startup.

CAUSE: The program checks itself for unexpected changes each time it starts up, as a precaution against viruses. If you see this message, you should run a virus checking program on your disk and re-install the program from the master diskettes (being careful to keep them write-protected). In most cases, you can continue and execute the program without major ill effects, but there is a risk that you will contaminate or corrupt other files in your system.

PROBLEM: Error message about "Unable to allocate memory or bad format for module XXXX".

CAUSE: This message can appear for one of the following reasons:

- The module in question is obsolete, i.e. it is part of an older DesignWorks release.
- The module file is corrupted.
- There was insufficient memory to load the module.
- The module was not authorized for operation with your copy of DesignWorks (in this case another message will precede this one).

Site or Network Licensing

When you purchase a copy of our software you are licensed to store and use it on a single computer at one time, making copies only for backup purposes. We have very attractive multiple unit licensing arrangements for organizations that are interested in using our software on a number of machines on the same network. Please contact us directly for more information.

Copy Protection

For your convenience, the DesignWorks disk is not copy protected in any way, allowing you to copy the program onto your hard disk or backup floppies as desired. This does **not** imply that you are allowed to make copies for use on more than one computer, in fact you are specifically prohibited from doing so by your license agreement. By respecting that agreement, you will help us continue to bring you useful software products at reasonable prices.

Technical Support

If you have any questions, problems, suggestions, congratulations or whatever, you can contact us by phone at (604) 522-6200 (9 - 5 Pacific time, please ask for DesignWorks Technical Support), by fax at (604) 522-3972, on AppleLink in the Third Parties folder, or by mail at:

Technical Support,
Capilano Computing,
960 Quayside Drive, Suite 406,
New Westminster, B. C.
Canada V3M 6G2

Bulletin Board Service

The DesignBoard BBS is available especially for our customers to access technical support, download updates and technical notes, exchange files and communicate with Capilano personnel.

Access number: (604) 522-4361

Access parameters: 2400 Baud, 8 bits, no parity

On your first access you can use basic functions and request validation, or you can phone us in advance and we will create an ID for you.

Installer Credits

This program is being distributed using the Apple Installer version 3.3 licensed from Apple Computer, Inc. and *StuffIt* installer technology licensed from Aladdin System. *StuffIt* is a file compression, encryption and file archival utility by Aladdin Systems, 217 E. 86th St., #153, NY, NY, USA.

Changes Since DesignWorks™ 3.0

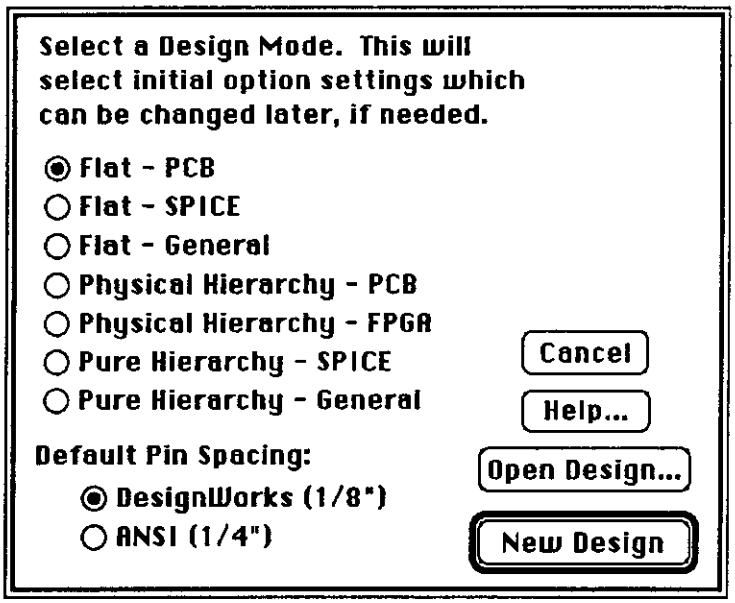
This section describes changes and additions to DesignWorks since the original 3.0 release. Also included are some corrections and supplementary information for the 3.0 manual provided with this package.

New Design Options

The most visible change in 3.1 is the addition of an options box which is displayed when the program starts up or when the New Design command is selected.

Note: If you always use the same options and want to override this box, new setup file options NEWDESIGN and NOUNTITLED allow you to control the creation of untitled designs. See more information on these options below.

This box does not actually implement any new options, it just groups them by application type and encourages you to select the appropriate mode in advance. All options can be set individually later in the design if desired, although this box can only be displayed for a new design.



Design Mode

The upper group of buttons allows you to choose the initial option settings for your design based on the type of application. The following table summarizes the usage of these modes.

Mode	Description
Flat - PCB	This mode is intended for creating small to medium designs intended to be implemented as a Printed Circuit Board and where hierarchical design is not required. This is the default mode and the best one to choose if you're not sure.
Flat - SPICE	This mode is intended for creating small to medium designs for input to a SPICE-based analog simulator. SPICE-format names will be assigned automatically.
Flat - General	This mode enables a minimum of options. It is used for block diagramming, simulation or analysis where device packaging is not needed.
Physical Hierarchy - PCB	This mode is intended for creating larger PCB designs with hierarchy.
Physical Hierarchy - FPGA	This mode is for hierarchical designs which require physical device data (i.e. for simulation or physical layout information), but do not need automatic gate assignment or name display.
Pure Hierarchy - SPICE	This mode is intended for creating hierarchical circuits (i.e. with SUBCKT definitions) for use with a SPICE-based simulator.
Pure Hierarchy - General	This mode is intended for general hierarchical design. NOTE: Pure hierarchy mode cannot be used with the DesignWorks Digital Simulator and is not recommended for PCB designs.

NOTE: It is strongly recommended that you read Chapter VII - Hierarchical Design in the DesignWorks manual before using any of the non-Flat modes.

The following table provides detail on the DesignWorks options enabled in each design mode. These options can be individually changed at any time during the design process, although it is usually best to select the desired options before a large amount of schematic entry is completed. This avoids making manual changes to name assignments, etc. later in the process.

Button	Hier. Mode	Dev Auto Vis	Packaging	Text Tool Sets	Name Prefix Field	Unit and Part Vis	Spice Vis	Value Vis
Flat - PCB	Flat	Off*	On	Name	Name.Prefix	On	Off	On
Flat - SPICE	Flat	On	Off	Name	Name.Spice	Off	On	Off
Flat - General	Flat	Off	Off	Name	Name.Prefix	Off	Off	On
Physical Hierarchy - PCB	Physical	Off	On	InstName	Name.Prefix	On	Off	On
Physical Hierarchy - FPGA	Physical	Off	Off	Name	Name.Prefix	Off	Off	On
Pure Hierarchy - SPICE	Pure	On	Off	Name	Name.Spice	Off	On	Off
Pure Hierarchy - General	Pure	Off	Off	Name	Name.Prefix	Off	Off	On

* Note: In Flat hierarchy mode, the Packaging option overrides the auto-naming feature in any case, so this setting is irrelevant. The Packager will place visible names on devices as they are placed.

In this table:

- Hier. Mode - refers to the Hierarchy Mode setting in the Design Preferences menu command.
- Dev. Auto Vis - refers to the Device Auto Naming - Visible option in the Design Preferences command. If this is On, the auto-assigned Name value will be displayed on the schematic.
- Packaging - refers to the Auto-Packaging Enabled menu command. If this is On, the device name and unit will be assigned automatically as each device is placed.

• Text Tool Sets - refers to the Text Tool Options item in Design Preferences. This determines which attribute field is set when the text tool is used to edit a device name on the schematic.
• Name Prefix Field - refers to the contents of the PrefixField design attribute field, set using the Set Design Attributes command. This determines where DesignWorks looks for a prefix to use in generating a device name.
• Unit, Part, Spice, Value Vis. - refer to the Visible By Default settings of the Unit, Part, Spice and Value attribute fields. These can be set using the Define Attribute Fields command. If this is On, the field value will be displayed on the schematic when the device is placed.

Default Pin Spacing

The lower group of buttons allows you to select from one of two commonly-used symbol styles supported by DesignWorks.

NOTE: This setting has no effect on the contents or availability of symbol libraries. It is up to you to choose the appropriate setting for the symbols you intend to use. The default DesignWorks installation installs only the DesignWorks-format symbols (1/8" pin spacing).

The following table summarizes the effect of these buttons.

DesignWorks (1/8")	This mode sets the default text sizes to 9-point for attribute and pin number text and the default pin spacing (i.e. for breakout symbols and DevEditor operations) to 2 grid units (approx. 1/8"). These settings will give best results with the DesignWorks-format symbol libraries provided with the package.
ANSI (1/4")	This mode sets the default text sizes to 12-point for attribute and pin number text and the default pin spacing to 4 grid units.* These settings will give correct text and line space proportions for ANSI-standard drawings when used with the ANSI or IEEE-format symbols provided. NOTE: The ANSI and IEEE symbols are not installed by the Easy Install option. You must select the Customize option to load them onto your disk.

* NOTE: The ANSI setting actually uses the setting specified in the PINSPACE keyword in the DesignWorks Setup file. If no PINSPACE item appears in the Setup File, the default value of 4 is used.

New Setup File Options

The new setup file options described in this section provide the ability to set default options in new designs. They are not required in normal usage. See the DesignWorks 3.0 User's Manual for more information on editing the setup file.

Adding Default Attribute Field Definitions

The setup file keyword ATTRFIELD allows you to specify attribute field definitions that will be included in every new design that is created. The format of this keyword is as follows:

```
ATTRFIELD fieldName;
```

or

```
ATTRFIELD fieldName/options;
```

"fieldName" can be any valid attribute field name, i.e. consisting of letters, digits and the period and underscore characters "." and "_". The maximum field name length is 16 characters and spaces are not allowed.

If no options are specified, the field will be a device field with default maximum length and switch settings.

A number of switch setting options can be specified as letters and digits after a "/" following the field name. If any digits follow the "/", they are taken to be the maximum data length for this field. The default is 32000. Any combination of other option letters can be selected from the following table. The order of the option letters is not important.

Any number of new fields can be added by using multiple ATTRFIELD lines. Any line specifying the name of an existing field will be ignored.

Option Letter	Meaning
A	Allow Carriage Return character in data
C	Allowed in design
D	Allowed in devices
F	Fixed, i.e. user cannot remove field definition
G	Group with Name
I	Instance data, i.e. can have different value inside different instances of the same hierarchical block
L	Location fixed on schematic

N	Only allow name characters in field value (NOTE: This is not enforced in the current version of DesignWorks)
P	Allowed in pins
R	Rotate with object
S	Allowed in signals
T	Do not display value list, even if a .List field is present
U	Only allow numeric characters in field value (NOTE: This is not enforced in the current version of DesignWorks)
V	Visible by default
W	Show Field Name by default
X	Read only value
Y	In Primary list by default

ATTRFIELD examples:

```
ATTRFIELD Aperture/SY10;
ATTRFIELD InputDelay/PYU10;
ATTRFIELD ModelText;
```

Setting New Design Command Action

The NEWDESIGN setup file keyword allows you to control the function of the New Design menu command. If this keyword does not appear in the setup file, the New Design command will display the mode selection box described earlier in these notes.

NOTE: If you hold the Option key pressed on the keyboard while selecting the New Design command, the New Design mode box will be displayed regardless of the settings in the NEWDESIGN setup file keyword.

The NEWDESIGN keyword can be followed by arguments in one of the following two forms:

```
NEWDESIGN designMode pinMode;
```

or

```
NEWDESIGN fileName;
```

In the first form, NEWDESIGN is followed by two integers, separated by blanks. These numbers in effect select the option buttons in the New Design mode selection dialog. The first integer can be from 1 to 7 and selects one of the modes described in the New Design command above in the order that the buttons appear in the box. I.e. 1 = Flat - PCB and 7 = Pure

Hierarchy - General. The second integer selects the pin space mode and can be 1 (DesignWorks 1/8") or 2 (ANSI 1/4").

The second form of the NEWDESIGN keyword allows you to specify the name of a file to open when the program is started or the New Design command is selected. This file can be an empty template containing your standard attribute fields and sheet settings. The format for the file name is the same as described for the DESIGN keyword in the DesignWorks User's Guide. This keyword is equivalent to performing an Open on the file except that the window will be titled "Designxxx" and the Save command will be disabled to avoid accidentally overwriting the template file.

NOTES: The name of the file specified after NEWDESIGN cannot start with a digit or the first form will be assumed.

NEWDESIGN examples:

```
NEWDESIGN 2 1; { Flat - SPICE }
NEWDESIGN :Template Files:My Startup Design;
```

Disabling Untitled Design at Startup

A new setup file keyword controls the creation of a new, untitled design when the program is first started. If the NOUNTITLED keyword appears in the file, the program will start up and just display its menus, with no default window opened. This entry will appear as follows:

```
NOUNTITLED;
```

Fine-Tuning Pin Number Text Display

The PINTEXT keyword allows you to adjust the display position of pin numbers on devices.

IMPORTANT: This adjustment should not be required in normal use and should be used with caution. No checking is done on the range of these settings.

The format of this keyword is as follows:

```
PINTEXT offsetH offsetV;
```

"offsetH" and "offsetY" are positive or negative integer offsets which adjust the text position by the given amount in screen dots at Normal Size screen scale. All devices in the design are equally affected.

NOTE: Changing these numbers in the setup file will not automatically recalculate the positions of pin numbers in existing designs. You can force a recalculate by using the Design Preferences command to change the pin text font or size, then change it back to the original setting.

Disabling "Loose End" Markers on Signal Lines

The NOLOOSEENDS setup file keyword disables the cross-markers that are normally displayed on the screen at the ends of unconnected line segments. The format of the command is:

```
NOLOOSEENDS;
```

Physical Hierarchy Mode Changes

A number of small refinements have been made in the operation of the hierarchy mode setting and device naming and packaging features. These changes are primarily to simplify the creation of hierarchical PCB designs. These changes include:

- If the hierarchy mode is changed in the Design Preferences command, an option box is now displayed with options and suggestions for reformatting the design. For example, if switching between Flat and Physical mode, the device Name field should be copied to the InstName field in most cases.

- If the InstName field is changed and Name has no value, or has a default value and is invisible, then copy the new InstName value to Name. This ensures a minimum of surprises when reference is made to the Name field by a netlist or other means.

- If Name is not visible, use Name.Pt to display InstName field. The standard libraries provided with DesignWorks specify a default position for the Name field in the Name.Pt field. However, in Physical mode, the device name is usually stored in InstName. This change ensures that names are positioned as expected.

- Improvements in auto-positioning of attributes, especially for grouped and rotated items

- The Packager is now used to perform automatic name assignment. This results in more "intelligent" generation of names.

- On a Paste or Duplicate, new names are assigned to devices even if they already have them.

• A new text tool option allows you to select when the pencil tool sets the Name or InstName attribute field. In Physical Hierarchy designs intended for PCB use, the device name is usually stored in the InstName field. This option can be set using the Design Preferences command.

• To provide visual feedback on which attribute field is being edited, the name of the current field is shown in the status box at the lower left of the schematic window.

• Simulation I/O devices are considered to be pseudo-devs, i.e. not named automatically.

Report Enhancements

Several new report form keywords have been added to address specific netlist applications.

$ASSIGNINSTNAMES

This works exactly like $ASSIGNNAMES except that it sets the InstName attribute field instead of the Name field. This is used for creating netlists from designs created in Physical Hierarchy mode.

$COUNTINST - after $DEVICES or $SIGNALS

COUNTINST works exactly like $COUNT except that it counts the number of devices with different InstName values, instead of Name values.

$LEXICAL/$NUMERIC - After $SORT

These keywords control the way items are sorted in a netlist. The default method, which matches all previous DesignWorks versions, is $NUMERIC. In this mode, any numeric portion of a text string is extracted and treated as an integer. For example, this means that U10 will appear after U9 in a component list. The disadvantage of this method is that U010 and U10 will be treated as the same name, or component values .001 and .01 will be combined (because "001" and "01" have the same integer value). The $LEXICAL option performs a purely lexical sort, i.e. by ASCII code. E.g. U1, U11, U100, U2, U22, U3, etc. This is intended when sorting by component values or other random data, but will not normally give the desired results for names.

The following example shows the usage of this keyword.

```
$SORT $DEVICES &Part $LEXICAL &Value
```

This example will sort devices by Part field using the default numerical method, then sort any devices with the same Part value by the string in the Value field using the lexical method.

$ASSIGNNAMES - Functional Change

ASSIGNNAMES now reassigns all temporary names that were assigned by any previous report.

Library Enhancements

New Symbol Format

Due to popular demand, a set of compact component symbols has been added to the DesignWorks libraries. These symbols do not meet ANSI drawing standards, but allow much more schematic to be placed on a sheet. These symbols are now the default installation. See the Installation section earlier in these notes for more information.

3.0 Manual Errata

Selecting Background Objects

Page 83 in the DesignWorks/Schematic Reference Manual indicates that the Command and Shift keys are used to select background objects. This should read Command and Option.

This page is intentionally left blank.

DesignWorks™ 3.1.1 Release and Installation Notes

These notes provide supplementary information on DesignWorks version 3.1.1, released June 10, 1993.

This version fixes some problems reported in the 3.1 release and adds some minor new functions, primarily for operation with the new DesignWorks Simulator.

> NOTE: These notes completely replace the DesignWorks 3.1 Release and Installation Notes
>
> See the section later in these notes for information on the changes in the 3.1.1 version that are not covered by the DesignWorks/Schematic Version 3.0 Reference Manual included with this release.

Thank-you

Thank-you for purchasing the DesignWorks package from Capilano Computing. We are sure that you will find the package productive and enjoyable to use. Should you have any problems, questions, or comments, please contact us at the number given below.

Site or Network Licensing

When you purchase a copy of our software you are licensed to store and use it on a single computer at one time, making copies only for backup purposes. We have very attractive multiple unit licensing arrangements for organizations that are interested in using our software on a number of machines on the same network. Please contact us directly for more information.

©1993 Capilano Computing Systems Ltd. All rights reserved.

Copy Protection

For your convenience, the DesignWorks disk is not copy protected in any way, allowing you to copy the program onto your hard disk or backup floppies as desired. This does **not** imply that you are allowed to make copies for use on more than one computer, in fact you are specifically prohibited from doing so by your license agreement. By respecting that agreement, you will help us continue to bring you useful software products at reasonable prices.

What you have received

This package contains the 3.1.1 release of DesignWorks, and should contain the following:

1 Binder
1 License Agreement
1 Release Notes
7 Program Diskettes
1 Reference Manual Package
1 Registration Card
1 Serial Number Label or Sheet
1 Support Plan Enrollment Card

NOTE: Some of the above items will not be included in upgrade packages.

Installation Notes

See also: Chapter II - Getting Started in the DesignWorks/Schematic manual.

In order to simplify installation and reduce the number of diskettes required to distribute DesignWorks, the release files are compressed onto a small number of diskettes using the Apple Installer and the StuffIt™ utility.

The Installer allows you to install a complete basic system in one operation, or to choose specific groups of items to be installed.

WARNING: If you already have a DesignWorks 3.1.1 installed on your disk the Installer will completely remove and replace any folders in "DesignWorks™ 3.1.1" having the same names as folders in the new set. Move any files you want to keep to another location before installing the new software.

Program Installation Procedure

Before starting the installation procedure you should take the following preparatory steps:

1) The Installer requires that it be the only program running on the machine it is installing onto. Before starting the Installer, shut down all other applications.

NOTE: If you are using Macintosh System Software Version 7.0 or later, the installation process will be significantly faster and more trouble-free if you restart your machine without extensions.

This is done by selecting the Restart item in the Finder's Special menu, then holding the Shift key on the keyboard depressed until you see the message "Extensions Off". This will also disable virus detection software, as described in the next step.

2) If you have any virus detection software on your machine, you may find some interaction with the Installer. It is best to disable any virus software before proceeding.

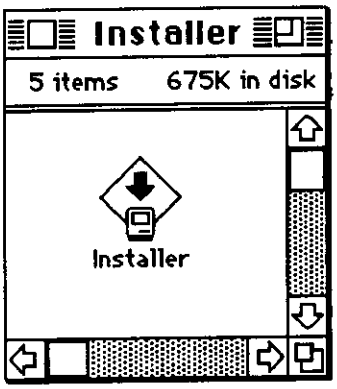
3) Depending on which disk you are installing on, the Installer may require that you disable File Sharing, or at least ensure that no folders are exported from your machine during the installation.

4) The standard installation of DesignWorks and its default libraries requires about 16 megabytes of disk space. Most of this space is taken up by parts libraries. If you want to install the default system, you will need to ensure that this amount of space is available. You can use the "Customize" option to choose only a basic program installation to get started, or to install additional libraries. In any case, a minimal installation requires at least 5 megabytes.

NOTE: The DesignWorks release includes 3 complete sets of libraries in different symbol formats. By default, only the DesignWorks (compact) symbols are installed. You can alternatively install the ANSI or IEEE symbol libraries. You cannot install the ANSI and Compact libraries in a single operation because of file name conflicts. If you want to install both, you will have to install one set, then move or rename the folder, then install the second set.

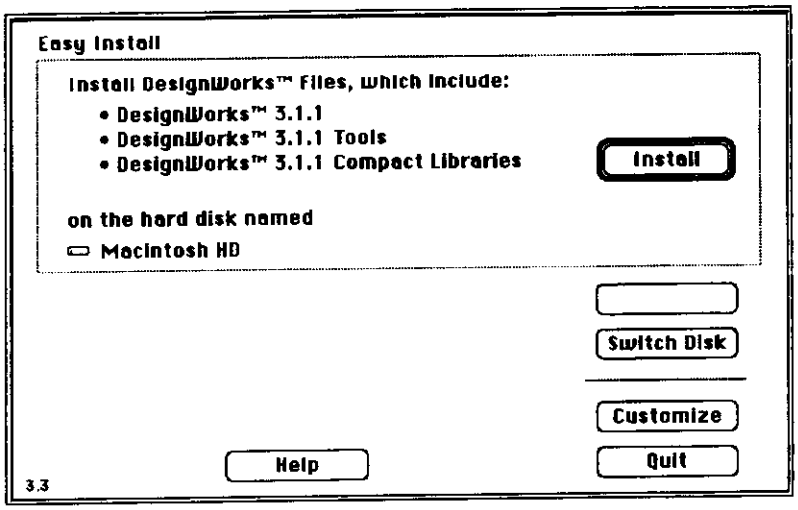
To start the installation:

1) Insert the diskette labeled "Installer". You should see the following files on the diskette:



2) Do not copy any of these files onto your hard disk! Double-click on the file "Installer". This will start up the installer program.

3) The following box will be displayed:



At this point you can choose one of two paths:

- Clicking the Install button will install the DesignWorks standard files on your disk.
- Clicking the Customize button will allow you to choose which library groups you install. If you are installing DesignWorks 3.1.1 for the first time, you may wish to install only the basic program set, since this is more than sufficient to get started with.

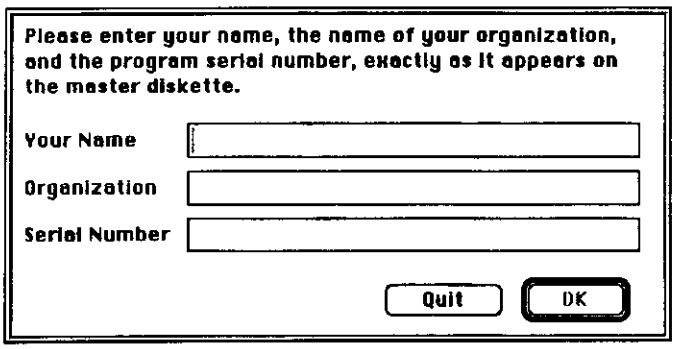
5) The installer will now copy the files from the installation disks, and then unpack the files onto your disk. It will ask for the diskettes it requires as it proceeds.

6) The program can now be started by clicking on the "DesignWorks™ 3.1.1" icon inside the DesignWorks 3.1.1 folder!

A basic set of libraries are included in the Basic Libs folder in the program installation. Before proceeding with a significant design, you should review the Guide to Libraries in the Notes section of the manual.

Serial Number Entry

The first time you start the program, you will see the following box:


Please enter your name, the name of your organization, and the program serial number, exactly as it appears on the master diskette.
Your Name
Organization
Serial Number
Quit
OK

Enter the following information:

- Your name and the name of your organization. This information will be displayed each time the program is started.
- The serial number as provided with your master diskette. This is a sequence of 20 letters and numbers. It can be entered either with or without separating spaces. All letters must be upper case.

After this information is entered once, it is kept with the program and you will not need to enter it again.

DesignWorks uses a network copy protection scheme that limits the number of users of a single copy on a network. The serial number contains encoded in it the number of authorized users and validation codes for add-on tools. You can upgrade your copy at any time by contacting Capilano Computing at the phone number given elsewhere in this manual.

Correcting or Re-entering the Serial Number

If for any reason you should wish to re-enter this information, you can hold the Option key pressed while starting the program from the Finder. The program will then redisplay the serial number entry box.

Troubleshooting Installation

PROBLEM: Program seems to run, but no windows appear and the Window and Tools menus have nothing in them, or tools are missing.

CAUSE: The program has failed to locate any of the tool files. This could be due to any of the following:

- The tools are not in the same folder as the DesignWorks™ 3.1.1 application or the Tools folder has been renamed.
- The DesignWorks Setup file has been renamed or is not in the same folder as the DesignWorks™ 3.1.1 application.
- The folder name specified in the TOOLFOLDER line of the DesignWorks Setup file is incorrect.

PROBLEM: Error message about "Could not open library XXXX"

CAUSE: The name or location of a library specified in the DesignWorks Setup file is incorrect.

PROBLEM: Error message about "Unexpected changes have been made to the application..." appears on startup.

CAUSE: The program checks itself for unexpected changes each time it starts up, as a precaution against viruses. If you see this message, you should run a virus checking program on your disk and re-install the program from the master diskettes (being careful to keep them write-protected). In most cases, you can continue and execute the program without major ill effects, but there is a risk that you will contaminate or corrupt other files in your system.

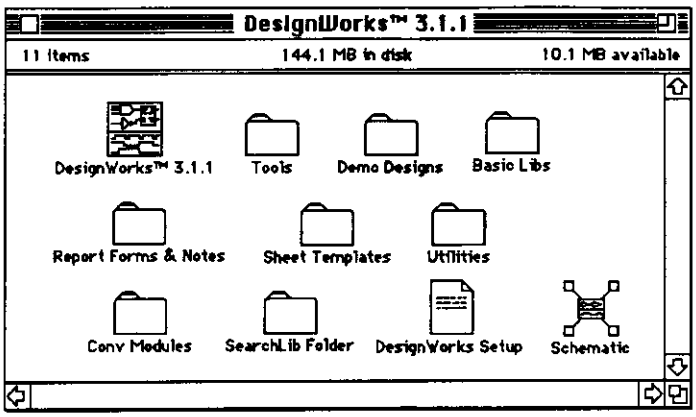
PROBLEM: Error message about "Unable to allocate memory or bad format for module XXXX".

CAUSE: This message can appear for one of the following reasons:

- The module in question is obsolete, i.e. it is part of an older DesignWorks release.
- The module file is corrupted.
- There was insufficient memory to load the module.
- The module was not authorized for operation with your copy of DesignWorks (in this case another message will precede this one).

Files Installed

The following Finder window shows a typical DesignWorks installation:



DesignWorks™ 3.1.1	This is the program itself. Double-click on this icon to start it up. This program is only a "shell" which loads the Schematic file and other tools.
Demo Designs	This is a folder containing some sample design files.
DesignWorks Setup	This is a text file which determines where DesignWorks searches for modules and libraries, default colors for objects, an initial design file to open, and other parameters. These are described in Appendix D of the DesignWorks manual. The basic setup file provided assumes the existence of a folder called Tools containing at least the "LibIO" module, and a folder called "Basic Libs" containing libraries.
Tools	This is a folder containing any code modules that are to be loaded when DesignWorks starts up. The name of this folder is specified in the Setup file. If the folder name is changed, as corresponding change must be made in the Setup file or no modules will be found. The only module that is essential for normal DesignWorks operation is "LibIO", which handles the Parts palette and all device library operations.
Sheet Templates	This folder contains template files for standard sheet sizes. These can be imported to your design using the Import Sheet Info command.

Report Forms & Notes	This is a folder containing the "form" files and circuit examples for the various standard report formats produced by the Report module. Information on these formats is given in the Report Generator manual and in text comments in the form files themselves.
Basic Libs	Any library files in this folder will be opened automatically when the program is started.
Libs	This folder contains all libraries installed by the Installer program. These libraries are not opened automatically at startup, but can be opened using the Open Lib command.
Utilities	This folder contains a number of utility tools and associated documentation.
Conv Tools	This folder contains the tools needed for conversion of pre-3.0 DesignWorks files to the new format. See the notes below before proceeding.
SearchLib Folder	This folder contains items for the SearchLib tool, which is used to locate parts in the libraries by description.
Schematic	This is the main schematic drawing tool for DesignWorks. It is loaded automatically when DesignWorks starts up. The program should normally be started by double-clicking on the DesignWorks™ 3.1.1 icon rather than on this file.

If you are updating from an older version of DesignWorks

Updating From 1.x or 2.x

If this package is an upgrade for an existing DesignWorks installation, please note that a conversion procedure must be performed on any files created using version 2.5.6 or older. Instructions for this procedure are in the DesignWorks/Converter manual behind the Notes tab in the binder.

New Attribute Conversion Option

The Converter tool supplied with the DesignWorks 3.1.1 version has a new option to facilitate attribute field conversion. The Converter options box now contains the following additional switch:

Clicking on this switch will display a standard file open box. You can then select a "mapping file", i.e. a text file specifying the field names existing in the 2.x file and the desired new names.

This file must consist of text only (i.e. saved with TeachText, a programming text editor, or a word processor with a "text only" option). Exactly one pair of names (old name, then new name) must appear on each line, separated by one or more spaces. The "new name" item can be specified as $Delete, in which case all items having that field name will be deleted.

If the keyword $Delete appears by itself on a line, this instructs the program to delete all attributes that are not explicitly named in this file.

An example mapping file is provided on the 3.1.1 release disks.

Updating from 3.0 or 3.1

DesignWorks 3.0, 3.1 and 3.1.1 are fully file compatible, however an error existed in the packaging information for some of the library parts provided with the 3.0 release. This affected only the small number of parts which have multiple different symbols applicable to the same package. If you are continuing to use libraries supplied with 3.0 and this problem is a concern for you, please see the notes on the FixLib tool in the Utilities folder. The libraries provided with the full 3.1 or 3.1.1 releases are corrected.

Technical Support

If you have any questions, problems, suggestions, congratulations or whatever, you can contact us using any of the following methods:

Phone	(604) 522-6200 (9 - 5 Pacific time, please ask for DesignWorks Technical Support)
Fax	(604) 522-3972
AppleLink	D0816,
InterNet	D0816@applelink.apple.com
Mail	Technical Support, Capilano Computing, 960 Quayside Drive, Suite 406, New Westminster, B. C. CANADA V3M 6G2

Bulletin Board Services

DesignBoard BBS

The DesignBoard BBS is available especially for our customers to access technical support, download updates and technical notes, exchange files and communicate with Capilano personnel.

Access number: (604) 522-4361

Access parameters: 1200/2400/9600 Baud, 8 bits, no parity, 1 stop bit

On your first access you can use basic functions and request validation, or you can phone us in advance and we will create an ID for you.

AppleLink Capilano BBS

Users of the AppleLink network can access the Capilano bulletin board under the Third Party icon for the same services available on our DesignBoard service.

Installer Credits

This program is being distributed using the Apple Installer version 3.3 licensed from Apple Computer, Inc. and *StuffIt* installer technology licensed from Aladdin System. *StuffIt* is a file compression, encryption and file archival utility by Aladdin Systems, 217 E. 86th St., #153, NY, NY, USA.

Changes Since DesignWorks™ 3.0

This section describes changes and additions to DesignWorks since the original 3.0 release. Also included are some corrections and supplementary information for the 3.0 manual provided with this package.

Device Name Placement

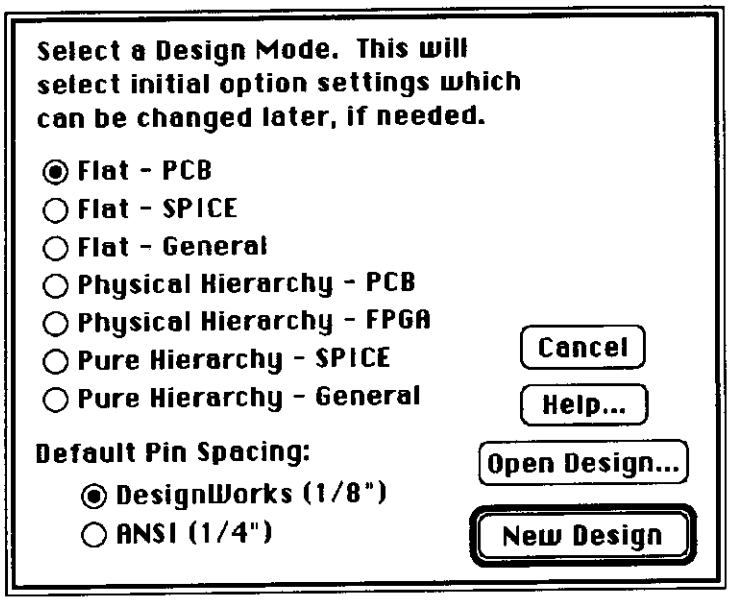
Clicking an releasing the text (pencil) tool on a device symbol without moving now causes the text to be positioned in the default location, rather than where the mouse button was released. If a default name offset in given in the Name.Pt attribute field, that position is used, otherwise a default position is computed by the program.

New Design Options

The most visible change in 3.1.1 is the addition of an options box which is displayed when the program starts up or when the New Design command is selected.

Note: If you always use the same options and want to override this box, new setup file options NEWDESIGN and NOUNTITLED allow you to control the creation of untitled designs. See more information on these options below.

This box does not actually implement any new options, it just groups them by application type and encourages you to select the appropriate mode in advance. All options can be set individually later in the design if desired, although this box can only be displayed for a new design.



Design Mode

The upper group of buttons allows you to choose the initial option settings for your design based on the type of application. The following table summarizes the usage of these modes.

Mode	Description
Flat - PCB	This mode is intended for creating small to medium designs intended to be implemented as a Printed Circuit Board and where hierarchical design is not required. This is the default mode and the best one to choose if you're not sure.
Flat - SPICE	This mode is intended for creating small to medium designs for input to a SPICE-based analog simulator. SPICE-format names will be assigned automatically.
Flat - General	This mode enables a minimum of options. It is used for block diagramming, simulation or analysis where device packaging is not needed.
Physical Hierarchy - PCB	This mode is intended for creating larger PCB designs with hierarchy.

Physical Hierarchy - FPGA	This mode is for hierarchical designs which require physical device data (i.e. for simulation or physical layout information), but do not need automatic gate assignment or name display.
Pure Hierarchy - SPICE	This mode is intended for creating hierarchical circuits (i.e. with SUBCKT definitions) for use with a SPICE-based simulator.
Pure Hierarchy - General	This mode is intended for general hierarchical design. NOTE: Pure hierarchy mode cannot be used with the DesignWorks Digital Simulator and is not recommended for PCB designs.

NOTE: It is strongly recommended that you read Chapter VII - Hierarchical Design in the DesignWorks manual before using any of the non-Flat modes.

The following table provides detail on the DesignWorks options enabled in each design mode. These options can be individually changed at any time during the design process, although it is usually best to select the desired options before a large amount of schematic entry is completed. This avoids making manual changes to name assignments, etc. later in the process.

Button	**Hier. Mode**	**Dev Auto Vis**	**Packaging**	**Text Tool Sets**	**Name Prefix Field**	**Unit and Part Vis**	**Spice Vis**	**Value Vis**
Flat - PCB	Flat	Off*	On	Name	Name.Prefix	On	Off	On
Flat - SPICE	Flat	On	Off	Name	Name.Spice	Off	On	Off
Flat - General	Flat	Off	Off	Name	Name.Prefix	Off	Off	On
Physical Hierarchy - PCB	Physical	Off	On	InstName	Name.Prefix	On	Off	On
Physical Hierarchy - FPGA	Physical	Off	Off	Name	Name.Prefix	Off	Off	On

Pure Hierarchy - SPICE	Pure	On	Off	Name	Name.Spice	Off	On	Off
Pure Hierarchy - General	Pure	Off	Off	Name	Name.Prefix	Off	Off	On

* Note: In Flat hierarchy mode, the Packaging option overrides the auto-naming feature in any case, so this setting is irrelevant. The Packager will place visible names on devices as they are placed.

In this table:

- Hier. Mode - refers to the Hierarchy Mode setting in the Design Preferences menu command.
- Dev. Auto Vis - refers to the Device Auto Naming - Visible option in the Design Preferences command. If this is On, the auto-assigned Name value will be displayed on the schematic.
- Packaging - refers to the Auto-Packaging Enabled menu command. If this is On, the device name and unit will be assigned automatically as each device is placed.
- Text Tool Sets - refers to the Text Tool Options item in Design Preferences. This determines which attribute field is set when the text tool is used to edit a device name on the schematic.
- Name Prefix Field - refers to the contents of the PrefixField design attribute field, set using the Set Design Attributes command. This determines where DesignWorks looks for a prefix to use in generating a device name.
- Unit, Part, Spice, Value Vis. - refer to the Visible By Default settings of the Unit, Part, Spice and Value attribute fields. These can be set using the Define Attribute Fields command. If this is On, the field value will be displayed on the schematic when the device is placed.

Default Pin Spacing

The lower group of buttons allows you to select from one of two commonly-used symbol styles supported by DesignWorks.

NOTE: This setting has no effect on the contents or availability of symbol libraries. It is up to you to choose the appropriate setting for the symbols you intend to use. The default DesignWorks installation installs only the DesignWorks-format symbols (1/8" pin spacing).

The following table summarizes the effect of these buttons.

DesignWorks (1/8")	This mode sets the default text sizes to 9-point for attribute and pin number text and the default pin spacing (i.e. for breakout symbols and DevEditor operations) to 2 grid units (approx. 1/8"). These settings will give best results with the DesignWorks-format symbol libraries provided with the package.
ANSI (1/4")	This mode sets the default text sizes to 12-point for attribute and pin number text and the default pin spacing to 4 grid units.* These settings will give correct text and line space proportions for ANSI-standard drawings when used with the ANSI or IEEE-format symbols provided. NOTE: The ANSI and IEEE symbols are not installed by the Easy Install option. You must select the Customize option to load them onto your disk.

* NOTE: The ANSI setting actually uses the setting specified in the PINSPACE keyword in the DesignWorks Setup file. If no PINSPACE item appears in the Setup File, the default value of 4 is used.

New Sub-Circuit Commands

A new sub-menu has been added under the Options menu in Schematic, entitled Sub-Circuit. Several of the commands in this menu are simply moved or renamed commands that existed in 3.0, but the "external sub-circuit" commands are new.

External Sub-Circuits

The concept of "external sub-circuits" is not new to this version, but it is much more visible due to the added menu commands. These commands were added primarily for use with the DesignWorks Simulator, although they also have other applications.

External vs. Internal Sub-Circuits

In order to simulate, analyze or generate reports from a hierarchical design, DesignWorks requires that all sub-circuits be loaded into memory. When a device symbol that represents a sub-circuit block is created in a library, the definition of it's sub-circuit can be associated with it in a number of ways:

- The sub-circuit can be stored with the symbol in the library, i.e. an "internal sub-circuit"

- The name and directory location of a circuit file can be stored with the symbol or added to the symbol after it is used in the schematic. The sub-circuit itself is not stored with the symbol, just the information required to locate it later. Schematic options allow the sub-circuit to be loaded automatically when the symbol is used, or on request at a later time. This is referred to as an "external sub-circuit".
- The symbol can be stored with no associated sub-circuit and one can be created or attached to it interactively after it has been placed in the schematic.

External References

The information used to locate an external sub-circuit is called an "external reference" and is stored in a number of attribute fields associated with the device. External references can be added to a device in various ways:

- When creating or editing a symbol in the DevEditor tool, an external reference is added by using the Sub-circuit and Part Type menu command and selecting the External Sub-Circuit option. This allows you to locate the external file, load the port definition and associate device pins with port connectors in the sub-circuit.
- If the Schematic tool is being used in conjunction with the DesignWorks Simulator, the SimLoad tool can be used to automatically locate and set external references for standard library parts as they are used or after they have been placed in a schematic.
- The Attach External Sub-circuit command in the Schematic tool can be used to associate an external sub-circuit with a symbol after it has been placed in a schematic.
- The external reference attributes can be entered manually. THIS IS NOT RECOMMENDED because any slight error in names or entry format will result in the file not being located.

See more information on sub-circuits and the SimLoad tool in the DesignWorks Simulator Reference Manual.

Creating External Sub-Circuits

External sub-circuits are simply normal DesignWorks design files with port connectors added to match the pins on the parent device. Following is a brief summary of the considerations affecting sub-circuits:

- In order to be loaded automatically using the SimLoad tool, the file must be named according to the conventions outlined in the DesignWorks Simulator manual.

• A port connector should be placed in the sub-circuit for each pin on the parent symbol, including no-connect pins. Failure to do this will result in the Schematic tool reporting an error each time the sub-circuit is opened and closed.

• Each port connector must be named to match the corresponding pin name on the parent device.

• The type of each port connector should match the type of the corresponding pin on the parent device. E.g. An input pin on the parent device should match with a "Port Conn In", or other appropriate type.

• The sub-circuit can contain any combination of other devices, including nested sub-circuits to any depth. The only restriction is that the sub-circuit may not contain the parent type or any other type that would create a recursive nesting of blocks.

Other considerations relating to the creation of sub-circuits are discussed in Chapter VII - Hierarchical Design in the DesignWorks/Schematic Reference Manual and in Chapter IV - Creating a Schematic for Simulation in the DesignWorks Simulator manual.

Sub-Circuit Menu Commands

IMPORTANT: When you attach or detach device sub-circuits, you are in effect changing the definition of the device. This always affects all other devices of the same type in the same design. This applies to all the sub-circuit commands following and any other sub-circuit editing operation in DesignWorks. If you want to affect only a single device, you must first use the Make Unique Type command in the Part Type sub-menu to isolate it from others of the same type.

Attach Sub-circuit

This is identical to the Attach Internal command described in the DesignWorks/Schematic Reference Manual. It allows you to choose a design that is already open in another window to be attached to the selected device symbol.

Refer to Chapter IX - Schematic Menu Reference for details.

Detach Sub-circuit

This is identical to the Detach Internal command described in the DesignWorks/Schematic Reference Manual. It detaches the sub-circuit that is displayed in the current circuit window from its parent device symbol. The sub-circuit will then be considered to be a completely separate design and the parent device symbol will be redefined as having no sub-circuit.

NOTE: Attach Sub-circuit works from the top down, i.e. you select the parent device symbol and then locate the sub-circuit to attach to it. Detach Sub-circuit works from the bottom up, i.e. the sub-circuit to be detached must already be open in the current window.

Refer to Chapter IX - Schematic Menu Reference for details.

Discard Sub-circuit

This new command discards any sub-circuit associated with the device that is selected on the schematic (and all others of the same type). It is similar to Detach Sub-circuit in that the parent device symbol will be redefined as being empty.

WARNING: The sub-circuit is completely lost and this operation cannot be undone!

Attach External Sub-circuit

This new command allows you to associate an external design file on disk with a device symbol in a schematic. It operates only on a single selected device at a time, but always affects all other devices of the same type in the design. It will optionally set the external reference attributes only, or set the external reference and immediately load and attach the sub-circuit.

The Attach External Sub-circuit command is also a convenient way of checking the status of an external reference. It locates the external file and compares the file's modified date with the parent device symbol. The "up to date" status and any problems locating the file will be displayed in the status box. If the selected device has an external reference which is currently out of date, clicking the OK button will reload it.

Update External Sub-circuits

This command allows you to scan all or part of a design and load or update external sub-circuits. This affects only devices that already have an external reference, created by one of the methods described previously. If the external file cannot be located, or if the external file is older or the same age as the last update, nothing is done. A summary box will be displayed after execution is complete showing counts of loaded and unlocated files.

NOTE: This command can dramatically increase the memory space occupied by your design.

Clear External Sub-circuits

This command allows you to scan all or part of a design and delete the sub-circuits from all devices having an external reference. This affects only devices that already have an external reference, created by one of the methods described previously. Devices with sub-circuit blocks that were not loaded using an external reference will not be affected.

This command is intended primarily to unload simulation models that were loaded using the SimLoad tool after they are no longer required. This can save considerable storage space for the design.

WARNING: This command discards a lot of information and cannot be Undone!

Auto-load External Sub-circuits

This command sets or clears the "Auto Load" option. It is disabled by default. When this option is enabled, each part is checked for an external reference when it is selected from a library. If there is a reference and the file can be located, it is loaded and attached immediately. Enabling this option has no effect on parts already placed in the design.

NOTE: If you select a part in the Parts palette that has already been used elsewhere in the design, the existing definition is used and no sub-circuit will be loaded.

External Reference Attribute Fields

The external reference information is stored in a number of attribute fields associated with the device, as summarized in the following table. Manual

editing of these fields is not recommended, since any slight spelling or format error will result in the external file not being loaded.

ExtCctName	This field contains the name of the design file containing the external sub-circuit.
ExtCctPath	This field contains the "path name" of nested directories leading to the file. The first item in the string is always the disk name, followed by a colon, followed by the names of nested folders, separated by colons.
ExtCctDate	This field contains an integer value expressed in decimal representing 1/60's of a second since Jan. 1, 1904. If this field is empty, the reference will be considered to be out of date.

New Setup File Options

The new setup file options described in this section provide the ability to set default options in new designs. They are not required in normal usage. See the DesignWorks 3.0 User's Manual for more information on editing the setup file.

Adding Default Attribute Field Definitions

The setup file keyword ATTRFIELD allows you to specify attribute field definitions that will be included in every new design that is created. The format of this keyword is as follows:

ATTRFIELD *fieldName*;

or

ATTRFIELD fieldName/options;

"fieldName" can be any valid attribute field name, i.e. consisting of letters, digits and the period and underscore characters "." and "_". The maximum field name length is 16 characters and spaces are not allowed.

If no options are specified, the field will be a device field with default maximum length and switch settings.

A number of switch setting options can be specified as letters and digits after a "/" following the field name. If any digits follow the "/", they are taken to be the maximum data length for this field. The default is 32000.

Any combination of other option letters can be selected from the following table. The order of the option letters is not important.

Any number of new fields can be added by using multiple ATTRFIELD lines. Any line specifying the name of an existing field will be ignored.

Option Letter	Meaning
A	Allow Carriage Return character in data
C	Allowed in design
D	Allowed in devices
F	Fixed, i.e. user cannot remove field definition
G	Group with Name
I	Instance data, i.e. can have different value inside different instances of the same hierarchical block
L	Location fixed on schematic
N	Only allow name characters in field value (NOTE: This is not enforced in the current version of DesignWorks)
P	Allowed in pins
R	Rotate with object
S	Allowed in signals
T	Do not display value list, even if a .List field is present
U	Only allow numeric characters in field value (NOTE: This is not enforced in the current version of DesignWorks)
V	Visible by default
W	Show Field Name by default
X	Read only value
Y	In Primary list by default

ATTRFIELD examples:

```
ATTRFIELD  Aperture/SY10;
ATTRFIELD  InputDelay/PYU10;
ATTRFIELD ModelText;
```

Setting New Design Command Action

The NEWDESIGN setup file keyword allows you to control the function of the New Design menu command. If this keyword does not appear in the

setup file, the New Design command will display the mode selection box described earlier in these notes.

NOTE: If you hold the Option key pressed on the keyboard while selecting the New Design command, the New Design mode box will be displayed regardless of the settings in the NEWDESIGN setup file keyword.

The NEWDESIGN keyword can be followed by arguments in one of the following two forms:

```
NEWDESIGN designMode pinMode;
```

or

```
NEWDESIGN fileName;
```

In the first form, NEWDESIGN is followed by two integers, separated by blanks. These numbers in effect select the option buttons in the New Design mode selection dialog. The first integer can be from 1 to 7 and selects one of the modes described in the New Design command above in the order that the buttons appear in the box. I.e. 1 = Flat - PCB and 7 = Pure Hierarchy - General. The second integer selects the pin space mode and can be 1 (DesignWorks 1/8") or 2 (ANSI 1/4").

The second form of the NEWDESIGN keyword allows you to specify the name of a file to open when the program is started or the New Design command is selected. This file can be an empty template containing your standard attribute fields and sheet settings. The format for the file name is the same as described for the DESIGN keyword in the DesignWorks User's Guide. This keyword is equivalent to performing an Open on the file except that the window will be titled "Designxxx" and the Save command will be disabled to avoid accidentally overwriting the template file.

NOTES: The name of the file specified after NEWDESIGN cannot start with a digit or the first form will be assumed.

NEWDESIGN examples:

```
NEWDESIGN 2 1; { Flat - SPICE }
NEWDESIGN :Template Files:My Startup Design;
```

Disabling Untitled Design at Startup

A new setup file keyword controls the creation of a new, untitled design when the program is first started. If the NOUNTITLED keyword appears in the file, the program will start up and just display its menus, with no default window opened. This entry will appear as follows:

NOUNTITLED;

Fine-Tuning Pin Number Text Display

The PINTEXT keyword allows you to adjust the display position of pin numbers on devices.

> IMPORTANT: This adjustment should not be required in normal use and should be used with caution. No checking is done on the range of these settings.

The format of this keyword is as follows:

PINTEXT *offsetH offsetV*;

"offsetH" and "offsetY" are positive or negative integer offsets which adjust the text position by the given amount in screen dots at Normal Size screen scale. All devices in the design are equally affected.

> NOTE: Changing these numbers in the setup file will not automatically recalculate the positions of pin numbers in existing designs. You can force a recalculate by using the Design Preferences command to change the pin text font or size, then change it back to the original setting.

Disabling "Loose End" Markers on Signal Lines

The NOLOOSEENDS setup file keyword disables the cross-markers that are normally displayed on the screen at the ends of unconnected line segments. The format of the command is:

NOLOOSEENDS;

Physical Hierarchy Mode Changes

A number of small refinements have been made in the operation of the hierarchy mode setting and device naming and packaging features. These

changes are primarily to simplify the creation of hierarchical PCB designs. These changes include:

- If the hierarchy mode is changed in the Design Preferences command, an option box is now displayed with options and suggestions for reformatting the design. For example, if switching between Flat and Physical mode, the device Name field should be copied to the InstName field in most cases.

- If the InstName field is changed and Name has no value, or has a default value and is invisible, then copy the new InstName value to Name. This ensures a minimum of surprises when reference is made to the Name field by a netlist or other means.

- If Name is not visible, use Name.Pt to display InstName field. The standard libraries provided with DesignWorks specify a default position for the Name field in the Name.Pt field. However, in Physical mode, the device name is usually stored in InstName. This change ensures that names are positioned as expected.

- Improvements in auto-positioning of attributes, especially for grouped and rotated items

- The Packager is now used to perform automatic name assignment. This results in more "intelligent" generation of names.

- On a Paste or Duplicate, new names are assigned to devices even if they already have them.

- A new text tool option allows you to select when the pencil tool sets the Name or InstName attribute field. In Physical Hierarchy designs intended for PCB use, the device name is usually stored in the InstName field. This option can be set using the Design Preferences command.

- To provide visual feedback on which attribute field is being edited, the name of the current field is shown in the status box at the lower left of the schematic window.

- Simulation I/O devices are considered to be pseudo-devs, i.e. not named automatically.

Report Enhancements

Several new report form keywords have been added to address specific netlist applications.

$COUNTINST - after $DEVICES or $SIGNALS

COUNTINST works exactly like $COUNT except that it counts the number of devices with different InstName values, instead of Name values.

$LEXICAL/$NUMERIC - After $SORT

These keywords control the way items are sorted in a netlist. The default method, which matches all previous DesignWorks versions, is $NUMERIC. In this mode, any numeric portion of a text string is extracted and treated as an integer. For example, this means that U10 will appear after U9 in a component list. The disadvantage of this method is that U010 and U10 will be treated as the same name, or component values .001 and .01 will be combined (because "001" and "01" have the same integer value). The $LEXICAL option performs a purely lexical sort, i.e. by ASCII code. E.g. U1, U11, U100, U2, U22, U3, etc. This is intended when sorting by component values or other random data, but will not normally give the desired results for names.

The following example shows the usage of this keyword.

```
$SORT $DEVICES &Part $LEXICAL &Value
```

This example will sort devices by Part field using the default numerical method, then sort any devices with the same Part value by the string in the Value field using the lexical method.

$REASSIGN - after $ASSIGNNAMES or $ASSIGNINSTNAMES

ASSIGNNAMES and ASSIGNINSTNAMES normally assign default names only to signals or devices that were unnamed, i.e. no name has been assigned by the user, the auto-naming features in Schematic or by any previous report.

However, in some cases, especially hierarchical designs, it is likely that duplicate names exists due to duplication of hierarchical blocks. For this reason, the $REASSIGN keyword has been added. When it appears anywhere after an ASSIGNNAMES and ASSIGNINSTNAMES, all names

that were auto-assigned by either the Schematic or the Report tool and have not been edited by the user will be reassigned a unique name.

The $REASSIGN keyword has been added to all ASSIGNINSTNAME commands in the standard report forms provided with DesignWorks. It will normally not be required in Flat or Pure mode designs.

$ASSIGNINSTNAMES

This works exactly like $ASSIGNNAMES except that it sets the InstName attribute field instead of the Name field. This is used for creating netlists from designs created in Physical Hierarchy mode.

Library Enhancements

New Symbol Format

Due to popular demand, a set of compact component symbols has been added to the DesignWorks libraries. These symbols do not meet ANSI drawing standards, but allow much more schematic to be placed on a sheet. These symbols are now the default installation. See the Installation section earlier in these notes for more information.

3.0 Manual Errata

Selecting Background Objects

Page 83 in the DesignWorks/Schematic Reference Manual indicates that the Command and Shift keys are used to select background objects. This should read Command and Option.

Index to DesignWorks 3.1.1 Release and Installation Notes

ASSIGNINSTNAMES report keyword 26, 27
ASSIGNNAMES report keyword 26
Attach External Sub-circuit command 17, 19
Attach Sub-circuit command 18
ATTRFIELD setup file keyword 21
Auto-load External Sub-circuits command 20
background objects 27
Clear External Sub-circuits command 20
converting 9
COUNTINST report keyword 26
Detach Sub-circuit command 19
DevEditor tool 17
devices
    name position 12
Discard Sub-circuit command 19
ExtCctDate attribute field 21
ExtCctName attribute field 21
ExtCctPath attribute field 21
external sub-circuits 16
    creating 17
flat hierarchy mode 13
hierarchy mode 12, 24
installation 3
InstName attribute field 14, 26, 27
LEXICAL report keyword 26
libraries 27
    installation 4
Name attribute field 14, 26
Name.Prefix attribute field 14
Name.Pt attribute field 12, 25
names
    devices 12
    port connector 18
New Design command 22
NEWDESIGN setup file keyword 22
NOLOOSEENDS setup file keyword 24
NOUNTITLED setup file keyword 24
NUMERIC report keyword 26
Packager tool 25
physical hierarchy mode 13, 24
pin spacing 15
PINTEXT setup file keyword 24
port connector 17, 18
pure hierarchy mode 14
REASSIGN report keyword 26
Report tool 26
    assigning names 26
    sorting 26
serial number 6
setup file 21
SimLoad tool 17
SORT report keyword 26
SPICE 13
sub-circuits 16
    external 16
    internal 16
symbol libraries 27
Update External Sub-circuits command 20

!!! IMPORTANT !!!

You Need This Sheet To Run Your Software!

Thank-you for purchasing this Capilano design software package. The Serial Number shown below must be entered when you first run the software on your machine, or if you are upgrading with new options. If you are running the software for the first time, you will be prompted to enter a serial number. If you are upgrading an existing package, hold the Option key while double-clicking on the application to get the serial number prompt.

Product Code ..... DWMP
Unit Number..... 272
Number of Users 1
Options ..... Simulator
Serial Number .....KYCW 8K73 61K4 3LJH CCCD

960 Quayside Drive, Suite 406,
New Westminster, B.C.
Canada V3M 6G2

(604) 522-6200
(604) 522-3972 Fax

DesignWorks™/Schematic 3.1.4 Update Notes

IMPORTANT NOTE: These notes are <u>supplemental</u> to the DesignWorks 3.1.1 Release and Installation Notes. DO NOT THROW THE 3.1.1 NOTES AWAY!

These notes provide information on DesignWorks/Schematic version 3.1.4 for users of version 3.1.1, 3.1.2 or 3.1.3. This update includes only the basic program installation as none of the files in the libraries section of the installation have been modified. In addition, these notes cover only the differences between this version and version 3.1.1. Users of version 3.1 or older should be performing a complete installation from the 8-disk program set to ensure that they have the latest libraries and documentation.

This version fixes some problems reported in the 3.1.1 release and adds the following new features:

- An auto-backup feature has been added to the Schematic tool. This will automatically create a backup copy of your design file each time a Save is done. NOTE: To use this feature you need to use the new DesignWorks Setup file supplied with this release, or modify your existing one. See below.
- The following new utility tools are included with every DesignWorks:
  - The Back Annotation tool, formerly sold as a separate product, is now included with DesignWorks, starting with this release. This tool will back annotate pin, gate and package swap information from several common PCB layout packages.
  - Updater is a new tool which allows automatic updating of device, pin or signal attributes in a design based on information in a text "database" file.
  - FNReader allows direct import of schematic files created by Data I/O Corporation's DASH schematic capture package.

©1994 Capilano Computing Systems Ltd. All rights reserved.

• A number of enhancements have been made to the Report tool's command language to support Verilog output and some other new netlist forms. These changes are not described in these notes. For more information on Report changes, please contact us for a copy of the technical note "Report Changes in DesignWorks 3.1.4".

• A demonstration version of the DesignWorks Simulator is included with the Schematic release. This can be used to simulate small circuits and preview the capabilities of the full Simulator option. The Simulator demo is not installed with the basic installation and requires selection of the "Custom" installation option, described below.

What you have received

This package contains an update to the 3.1.4 release of DesignWorks for owners of versions 3.1.1, 3.1.2 or 3.1.4, and should contain the following:

1 Release Notes
2 Program Diskettes

Technical Support

If you have any questions, problems, suggestions, congratulations or whatever, you can contact us using any of the following methods:

Phone	(604) 522-6200 (9 - 5 Pacific time, please ask for DesignWorks Technical Support)
Fax	(604) 522-3972
AppleLink	CAPILANO
InterNet	capilano_computing@wimsey.com
Mail	Technical Support, Capilano Computing, 960 Quayside Drive, Suite 406, New Westminster, B. C. CANADA V3M 6G2

Note Regarding Version Numbering

This version is being numbered 3.1.4 to re-synchronize numbering with releases of DesignWorks that were shipped by Douglas Electronics with

numbers 3.1.2 and 3.1.3. Neither of these versions were ever shipped directly by Capilano Computing.

Site or Network Licensing

When you purchase a copy of our software you are licensed to store and use it on a single computer at one time, making copies only for backup purposes. We have very attractive multiple unit licensing arrangements for organizations that are interested in using our software on a number of machines on the same network. Please contact us directly for more information.

Installation Notes

Installing the DesignWorks 3.1.4 Update

The DesignWorks/Schematic 3.1.4 release includes only a basic set of program and library files. Most of the libraries (i.e. the contents of the "Libs" folder in the 3.1.1 installation) have not been modified and are not included in this update.

To install the DesignWorks 3.1.4 update:

- Restart your computer without extensions. Assuming you are using Macintosh System Software version 7.0 or later, this is done by holding the SHIFT key pressed while selecting the Restart command in the Finder's Special menu. Hold the SHIFT key until you see the message "Extensions Off". This step will ensure that any file sharing or virus protection software you may have in your system does not interfere with the installation process. If you are using an older version of the Macintosh system, then disable any virus detection software you may have before proceeding as it may interfere with the installation process.
- Insert the diskette "DesignWorks 3.1.4 Update Disk 1" into your diskette drive and double-click on the "DW 3.1.4 Update" icon. Click Continue on the welcome box. Read the last-minute installation notes that are displayed at this point, then click the Continue button.
- The next step depends on whether you want to install the Simulator demo version or not.

To install:	Do the following:
The DesignWorks 3.1.4 update	Just click the Install button. This is the default installation.
The DesignWorks 3.1.4 update and the Simulator demo	Click the Custom button, hold the SHIFT key and select both the "DW 3.1.4 Update" and the "DW Simulator Demo" items, then click the Install button.
The Simulator demo into an existing DesignWorks 3.1.4 installation	Click the Custom button, select the "DW Simulator Demo" item in the list, then click the Install button. When the "Install software into folder" box appears, select the location of the existing 3.1.4 folder and enter its name (if it is not already correct). The existing folder will not be deleted, but the new items will be installed inside it.

This procedure will create a new DesignWorks 3.1.4 folder containing a basic installation of the new version. In order to complete the installation, you should drag the following items from your existing DesignWorks folder to the new one:

- The Libs folder, which contains all the standard DesignWorks libraries. These were not included in the update.
- Any library, design or report form files you have created.

There should not be any file compatibility issues between the 3.1.1, 3.1.2, 3.1.3 and 3.1.4 versions of DesignWorks, but we recommend that you keep your master disks for the older version on hand until you have verified that all your design files can be opened and edited successfully with the new version.

Removing the Simulator Demo

If you wish to remove the Simulator demonstration version that was installed using the instructions in the previous section, simply drag the following folders to the Trash or to another folder for safekeeping:

- "Sim Tools" in the DesignWorks folder.
- "Models" in the DesignWorks folder.
- "Simulator Demos" inside the Demo Designs folder in the DesignWorks folder.

New Features

This section describes changes and additions to DesignWorks since the 3.1.1/3.1.2/3.1.3 releases.

Auto-Backup of Design Files

The Schematic tool now has the ability to automatically create a backup copy of your design file each time a Save is done. To enable this feature, the line:

AUTOBACKUP;

must appear in the DesignWorks Setup file. The setup file provided with this release includes this line. If you are updating from an older release, you may wish to add this command to your file. The auto-backup feature will then be enabled the next time you start up the program.

With this feature enabled, the following actions are taken each time you select the Save command in the File menu (assuming you have a design called "*designName*" open):

- If a file named "*designName*.Backup" already exists in the same folder as the current design, it is deleted.
- The existing design file "*designName*" is renamed to "*designName*.Backup".
- The new design information is saved as "*designName*".

New Tools

Several powerful new utilities are now included with every copy of DesignWorks. These notes provide only an overview of these items. Full documentation on each module is provided in a "Read Me" file provided with each one. These tools are all located in the Utilities folder in the DesignWorks installation.

BackAnno - PCB Back Annotation

The PCB Back Annotation tool "BackAnno" allows device package assignments to be automatically updated from a "was-is" text file created by an external system. Formats supported include Cadnetix (SCICARDS),

Douglas, Racal-Redac RINF and XCAD. Pin and gate swaps, and component renaming are supported in flat or hierarchical designs.

Updater - General Purpose Attribute Annotation

A general attribute back annotation tool called "Updater" is also included. This tool reads a simple tab-separated text file format and will update arbitrary device, signal or pin attribute fields throughout a design. This can be used as a general mechanist for applying user-supplied part numbers, simulation information or other data from a text database.

> NOTE: Updater should not be used to perform PCB back annotation. Updater modifies only exactly the fields specified and cannot modify pin numbers, which are not stored as attributes. In contrast, the BackAnno module mentioned above will make appropriate modifications to pin numbers and packaging information in devices when gate and pin swap functions are performed.

FNReader - FutureNet™ File Import

The FNReader tool will import schematic or symbol library files created by the FutureNet (™ Data I/O Corporation) schematic capture package. Optional attribute field name mapping allows you to specify conversions between attribute fields used in the original schematic and new field names to be used in the DesignWorks schematic.

Report Enhancements

Several new report forms, including Verilog, FutureNet/DASH, and SMASH are included in the Report Forms and Notes folder, marked as "Experimental".

> IMPORTANT: These new formats have not been field tested and should not be relied upon. We value any feedback you can provide on the effectiveness of these formats in your environment. Feel free to contact us to determine if updated versions of these items are available.

DesignWorks™ Simulator 3.1.4 Update Notes

These notes provide information on DesignWorks Simulator version 3.1.4 for users of version 3.1.1, 3.1.2 or 3.1.3. This update includes only the basic simulator tools plus a small number of models that have been modified.

This version fixes some problems reported in the 3.1.1 release. No significant new features have been added.

What you have received

This package contains an update to the 3.1.4 release of the DesignWorks Simulator for owners of versions 3.1.1, 3.1.2 or 3.1.4, and should contain the following:

1 Release Notes

1 Program Diskette

IMPORTANT: This upgrade should only be installed in conjunction with an upgrade to the Schematic portion of DesignWorks. The upgraded Simulator tools may not work correctly with pre-3.1.4 versions of DesignWorks/Schematic.

Technical Support

If you have any questions, problems, suggestions, congratulations or whatever, you can contact us using any of the following methods:

Phone	(604) 522-6200 (9 - 5 Pacific time, please ask for DesignWorks Technical Support)
Fax	(604) 522-3972
AppleLink	CAPILANO
InterNet	capilano_computing@wimsey.com
Mail	Technical Support, Capilano Computing, 960 Quayside Drive, Suite 406, New Westminster, B. C. CANADA V3M 6G2

DesignWorks™ 3.1.1 Release and Installation Notes

These notes provide supplementary information on DesignWorks version 3.1.1, released June 10, 1993.

This version fixes some problems reported in the 3.1 release and adds some minor new functions, primarily for operation with the new DesignWorks Simulator.

> NOTE: These notes completely replace the DesignWorks 3.1 Release and Installation Notes
>
> See the section later in these notes for information on the changes in the 3.1.1 version that are not covered by the DesignWorks/Schematic Version 3.0 Reference Manual included with this release.

Thank-you

Thank-you for purchasing the DesignWorks package from Capilano Computing. We are sure that you will find the package productive and enjoyable to use. Should you have any problems, questions, or comments, please contact us at the number given below.

Site or Network Licensing

When you purchase a copy of our software you are licensed to store and use it on a single computer at one time, making copies only for backup purposes. We have very attractive multiple unit licensing arrangements for organizations that are interested in using our software on a number of machines on the same network. Please contact us directly for more information.

©1993 Capilano Computing Systems Ltd. All rights reserved.

Note Regarding Version Numbering

This version is being numbered 3.1.4 to re-synchronize numbering with releases of DesignWorks that were shipped by Douglas Electronics with numbers 3.1.2 and 3.1.3. Neither of these versions were ever shipped directly by Capilano Computing.

Installation Notes

The DesignWorks Simulator 3.1.4 update includes all the basic Simulator program and demo files, but only the small number of models that have been updated. Most of the models (i.e. the contents of the "Models" folder in the 3.1.1 installation) have not been modified and are not included in this update.

To install the DesignWorks Simulator 3.1.4 update:

- Restart your computer without extensions. Assuming you are using Macintosh System Software version 7.0 or later, this is done by holding the SHIFT key pressed while selecting the Restart command in the Finder's Special menu. Hold the SHIFT key until you see the message "Extensions Off". This step will ensure that any file sharing or virus protection software you may have in your system does not interfere with the installation process. If you are using an older version of the Macintosh system, then disable any virus detection software you may have before proceeding as it may interfere with the installation process.
- Insert the diskette "DesignWorks Simulator 3.1.4 Update Disk" into your diskette drive and double-click on the "DW 3.1.4 Sim Update" icon. Click Continue on the welcome box. Read the last-minute installation notes that are displayed at this point, then click the Continue button.
- Click on the Install button. A file save box will now appear. You should use this box to enter the name of the existing DesignWorks 3.1.4 folder and to select the disk or folder than contains the DesignWorks folder. If you specify any other name or location, a new folder will be created containing only the Simulator update items.
- Drag the Models folder (and any other files you may have created or modified) from your old DesignWorks 3.1.1 folder into the new DesignWorks 3.1.4 folder, or re-install them from the 3.1.1 disks.
- Open the new folder "Drag to Models Folder" and drag all the files it contains to the Models folder. They will replace some existing files.

WARNING: If you already have a DesignWorks 3.1.1 installed on your disk the Installer will completely remove and replace any folders in "DesignWorks™ 3.1.1" having the same names as folders in the new set. Move any files you want to keep to another location before installing the new software.

Program Installation Procedure

Before starting the installation procedure you should take the following preparatory steps:

1) The Installer requires that it be the only program running on the machine it is installing onto. Before starting the Installer, shut down all other applications.

NOTE: If you are using Macintosh System Software Version 7.0 or later, the installation process will be significantly faster and more trouble-free if you restart your machine without extensions.

This is done by selecting the Restart item in the Finder's Special menu, then holding the Shift key on the keyboard depressed until you see the message "Extensions Off". This will also disable virus detection software, as described in the next step.

2) If you have any virus detection software on your machine, you may find some interaction with the Installer. It is best to disable any virus software before proceeding.

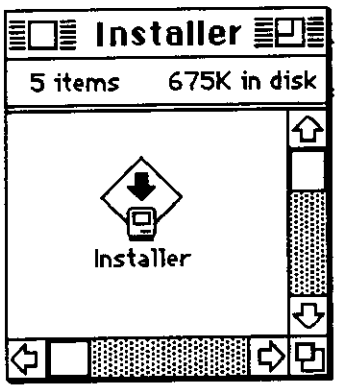
3) Depending on which disk you are installing on, the Installer may require that you disable File Sharing, or at least ensure that no folders are exported from your machine during the installation.

4) The standard installation of DesignWorks and its default libraries requires about 16 megabytes of disk space. Most of this space is taken up by parts libraries. If you want to install the default system, you will need to ensure that this amount of space is available. You can use the "Customize" option to choose only a basic program installation to get started, or to install additional libraries. In any case, a minimal installation requires at least 5 megabytes.

NOTE: The DesignWorks release includes 3 complete sets of libraries in different symbol formats. By default, only the DesignWorks (compact) symbols are installed. You can alternatively install the ANSI or IEEE symbol libraries. You cannot install the ANSI and Compact libraries in a single operation because of file name conflicts. If you want to install both, you will have to install one set, then move or rename the folder, then install the second set.

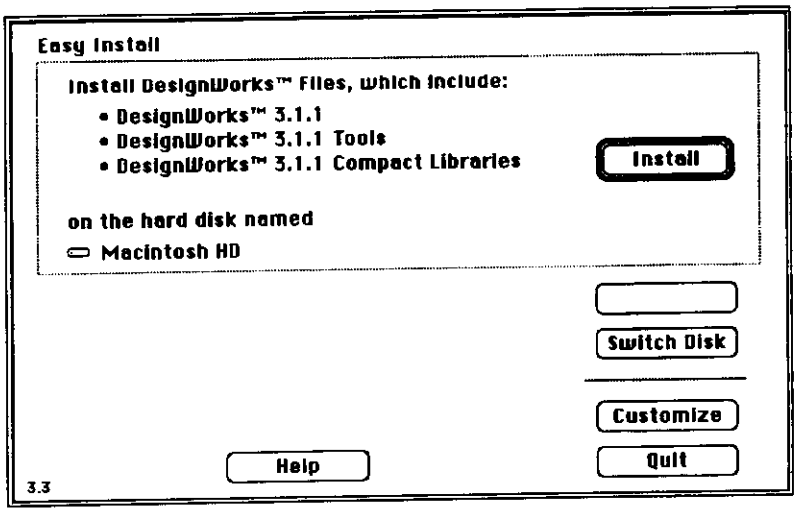
To start the installation:

1) Insert the diskette labeled "Installer". You should see the following files on the diskette:



2) Do not copy any of these files onto your hard disk! Double-click on the file "Installer". This will start up the installer program.

3) The following box will be displayed:



At this point you can choose one of two paths:

- Clicking the Install button will install the DesignWorks standard files on your disk.
- Clicking the Customize button will allow you to choose which library groups you install. If you are installing DesignWorks 3.1.1 for the first time, you may wish to install only the basic program set, since this is more than sufficient to get started with.

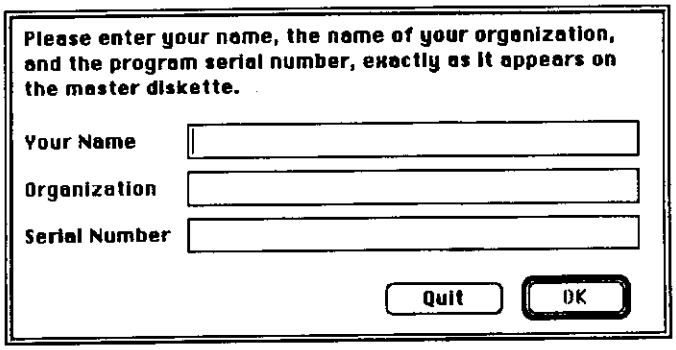
5) The installer will now copy the files from the installation disks, and then unpack the files onto your disk. It will ask for the diskettes it requires as it proceeds.

6) The program can now be started by clicking on the "DesignWorks™ 3.1.1" icon inside the DesignWorks 3.1.1 folder!

A basic set of libraries are included in the Basic Libs folder in the program installation. Before proceeding with a significant design, you should review the Guide to Libraries in the Notes section of the manual.

Serial Number Entry

The first time you start the program, you will see the following box:


Please enter your name, the name of your organization, and the program serial number, exactly as it appears on the master diskette.
Your Name
Organization
Serial Number
Quit
OK

Enter the following information:

- Your name and the name of your organization. This information will be displayed each time the program is started.
- The serial number as provided with your master diskette. This is a sequence of 20 letters and numbers. It can be entered either with or without separating spaces. All letters must be upper case.

After this information is entered once, it is kept with the program and you will not need to enter it again.

DesignWorks uses a network copy protection scheme that limits the number of users of a single copy on a network. The serial number contains encoded in it the number of authorized users and validation codes for add-on tools. You can upgrade your copy at any time by contacting Capilano Computing at the phone number given elsewhere in this manual.

Correcting or Re-entering the Serial Number

If for any reason you should wish to re-enter this information, you can hold the Option key pressed while starting the program from the Finder. The program will then redisplay the serial number entry box.

Troubleshooting Installation

PROBLEM: Program seems to run, but no windows appear and the Window and Tools menus have nothing in them, or tools are missing.

CAUSE: The program has failed to locate any of the tool files. This could be due to any of the following:

- The tools are not in the same folder as the DesignWorks™ 3.1.1 application or the Tools folder has been renamed.
- The DesignWorks Setup file has been renamed or is not in the same folder as the DesignWorks™ 3.1.1 application.
- The folder name specified in the TOOLFOLDER line of the DesignWorks Setup file is incorrect.

PROBLEM: Error message about "Could not open library XXXX"

CAUSE: The name or location of a library specified in the DesignWorks Setup file is incorrect.

PROBLEM: Error message about "Unexpected changes have been made to the application..." appears on startup.

CAUSE: The program checks itself for unexpected changes each time it starts up, as a precaution against viruses. If you see this message, you should run a virus checking program on your disk and re-install the program from the master diskettes (being careful to keep them write-protected). In most cases, you can continue and execute the program without major ill effects, but there is a risk that you will contaminate or corrupt other files in your system.

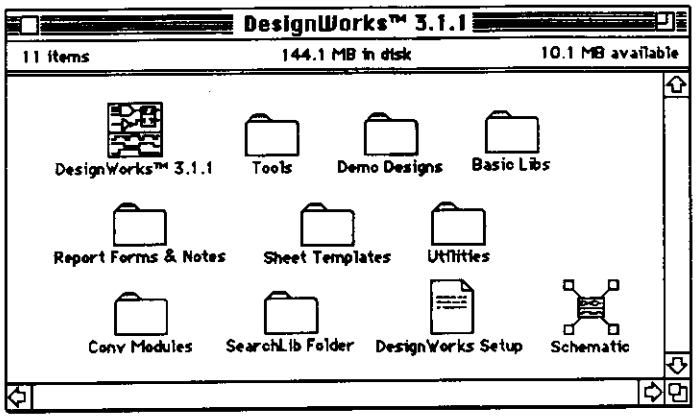
PROBLEM: Error message about "Unable to allocate memory or bad format for module XXXX".

CAUSE: This message can appear for one of the following reasons:

- The module in question is obsolete, i.e. it is part of an older DesignWorks release.
- The module file is corrupted.
- There was insufficient memory to load the module.
- The module was not authorized for operation with your copy of DesignWorks (in this case another message will precede this one).

Files Installed

The following Finder window shows a typical DesignWorks installation:



DesignWorks™ 3.1.1	This is the program itself. Double-click on this icon to start it up. This program is only a "shell" which loads the Schematic file and other tools.
Demo Designs	This is a folder containing some sample design files.
DesignWorks Setup	This is a text file which determines where DesignWorks searches for modules and libraries, default colors for objects, an initial design file to open, and other parameters. These are described in Appendix D of the DesignWorks manual. The basic setup file provided assumes the existence of a folder called Tools containing at least the "LibIO" module, and a folder called "Basic Libs" containing libraries.
Tools	This is a folder containing any code modules that are to be loaded when DesignWorks starts up. The name of this folder is specified in the Setup file. If the folder name is changed, as corresponding change must be made in the Setup file or no modules will be found. The only module that is essential for normal DesignWorks operation is "LibIO", which handles the Parts palette and all device library operations.
Sheet Templates	This folder contains template files for standard sheet sizes. These can be imported to your design using the Import Sheet Info command.

Report Forms & Notes	This is a folder containing the "form" files and circuit examples for the various standard report formats produced by the Report module. Information on these formats is given in the Report Generator manual and in text comments in the form files themselves.
Basic Libs	Any library files in this folder will be opened automatically when the program is started.
Libs	This folder contains all libraries installed by the Installer program. These libraries are not opened automatically at startup, but can be opened using the Open Lib command.
Utilities	This folder contains a number of utility tools and associated documentation.
Conv Tools	This folder contains the tools needed for conversion of pre-3.0 DesignWorks files to the new format. See the notes below before proceeding.
SearchLib Folder	This folder contains items for the SearchLib tool, which is used to locate parts in the libraries by description.
Schematic	This is the main schematic drawing tool for DesignWorks. It is loaded automatically when DesignWorks starts up. The program should normally be started by double-clicking on the DesignWorks™ 3.1.1 icon rather than on this file.

If you are updating from an older version of DesignWorks

Updating From 1.x or 2.x

If this package is an upgrade for an existing DesignWorks installation, please note that a conversion procedure must be performed on any files created using version 2.5.6 or older. Instructions for this procedure are in the DesignWorks/Converter manual behind the Notes tab in the binder.

New Attribute Conversion Option

The Converter tool supplied with the DesignWorks 3.1.1 version has a new option to facilitate attribute field conversion. The Converter options box now contains the following additional switch:

Clicking on this switch will display a standard file open box. You can then select a "mapping file", i.e. a text file specifying the field names existing in the 2.x file and the desired new names.

This file must consist of text only (i.e. saved with TeachText, a programming text editor, or a word processor with a "text only" option). Exactly one pair of names (old name, then new name) must appear on each line, separated by one or more spaces. The "new name" item can be specified as $Delete, in which case all items having that field name will be deleted.

If the keyword $Delete appears by itself on a line, this instructs the program to delete all attributes that are not explicitly named in this file.

An example mapping file is provided on the 3.1.1 release disks.

Updating from 3.0 or 3.1

DesignWorks 3.0, 3.1 and 3.1.1 are fully file compatible, however an error existed in the packaging information for some of the library parts provided with the 3.0 release. This affected only the small number of parts which have multiple different symbols applicable to the same package. If you are continuing to use libraries supplied with 3.0 and this problem is a concern for you, please see the notes on the FixLib tool in the Utilities folder. The libraries provided with the full 3.1 or 3.1.1 releases are corrected.

Technical Support

If you have any questions, problems, suggestions, congratulations or whatever, you can contact us using any of the following methods:

Phone	(604) 522-6200 (9 - 5 Pacific time, please ask for DesignWorks Technical Support)
Fax	(604) 522-3972
AppleLink	D0816,
InterNet	D0816@applelink.apple.com
Mail	Technical Support, Capilano Computing, 960 Quayside Drive, Suite 406, New Westminster, B. C. CANADA V3M 6G2

Bulletin Board Services

DesignBoard BBS

The DesignBoard BBS is available especially for our customers to access technical support, download updates and technical notes, exchange files and communicate with Capilano personnel.

Access number: (604) 522-4361

Access parameters: 1200/2400/9600 Baud, 8 bits, no parity, 1 stop bit

On your first access you can use basic functions and request validation, or you can phone us in advance and we will create an ID for you.

AppleLink Capilano BBS

Users of the AppleLink network can access the Capilano bulletin board under the Third Party icon for the same services available on our DesignBoard service.

Installer Credits

This program is being distributed using the Apple Installer version 3.3 licensed from Apple Computer, Inc. and *StuffIt* installer technology licensed from Aladdin System. *StuffIt* is a file compression, encryption and file archival utility by Aladdin Systems, 217 E. 86th St., #153, NY, NY, USA.

Changes Since DesignWorks™ 3.0

This section describes changes and additions to DesignWorks since the original 3.0 release. Also included are some corrections and supplementary information for the 3.0 manual provided with this package.

Device Name Placement

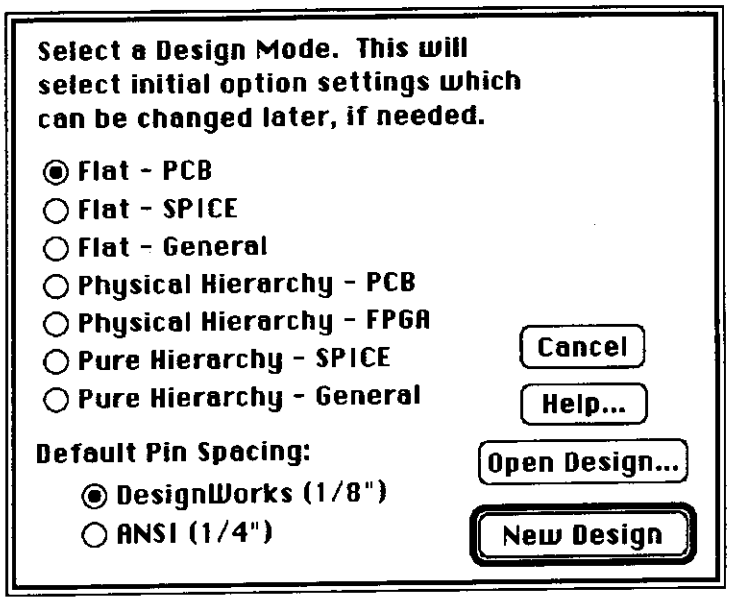
Clicking an releasing the text (pencil) tool on a device symbol without moving now causes the text to be positioned in the default location, rather than where the mouse button was released. If a default name offset in given in the Name.Pt attribute field, that position is used, otherwise a default position is computed by the program.

New Design Options

The most visible change in 3.1.1 is the addition of an options box which is displayed when the program starts up or when the New Design command is selected.

> *Note: If you always use the same options and want to override this box, new setup file options NEWDESIGN and NOUNTITLED allow you to control the creation of untitled designs. See more information on these options below.*

This box does not actually implement any new options, it just groups them by application type and encourages you to select the appropriate mode in advance. All options can be set individually later in the design if desired, although this box can only be displayed for a new design.



Design Mode

The upper group of buttons allows you to choose the initial option settings for your design based on the type of application. The following table summarizes the usage of these modes.

Mode	Description
Flat - PCB	This mode is intended for creating small to medium designs intended to be implemented as a Printed Circuit Board and where hierarchical design is not required. This is the default mode and the best one to choose if you're not sure.
Flat - SPICE	This mode is intended for creating small to medium designs for input to a SPICE-based analog simulator. SPICE-format names will be assigned automatically.
Flat - General	This mode enables a minimum of options. It is used for block diagramming, simulation or analysis where device packaging is not needed.
Physical Hierarchy - PCB	This mode is intended for creating larger PCB designs with hierarchy.

Physical Hierarchy - FPGA	This mode is for hierarchical designs which require physical device data (i.e. for simulation or physical layout information), but do not need automatic gate assignment or name display.
Pure Hierarchy - SPICE	This mode is intended for creating hierarchical circuits (i.e. with SUBCKT definitions) for use with a SPICE-based simulator.
Pure Hierarchy - General	This mode is intended for general hierarchical design. NOTE: Pure hierarchy mode cannot be used with the DesignWorks Digital Simulator and is not recommended for PCB designs.

NOTE: It is strongly recommended that you read Chapter VII - Hierarchical Design in the DesignWorks manual before using any of the non-Flat modes.

The following table provides detail on the DesignWorks options enabled in each design mode. These options can be individually changed at any time during the design process, although it is usually best to select the desired options before a large amount of schematic entry is completed. This avoids making manual changes to name assignments, etc. later in the process.

Button	**Hier. Mode**	**Dev Auto Vis**	**Packaging**	**Text Tool Sets**	**Name Prefix Field**	**Unit and Part Vis**	**Spice Vis**	**Value Vis**
Flat - PCB	Flat	Off*	On	Name	Name.Prefix	On	Off	On
Flat - SPICE	Flat	On	Off	Name	Name.Spice	Off	On	Off
Flat - General	Flat	Off	Off	Name	Name.Prefix	Off	Off	On
Physical Hierarchy - PCB	Physical	Off	On	InstName	Name.Prefix	On	Off	On
Physical Hierarchy - FPGA	Physical	Off	Off	Name	Name.Prefix	Off	Off	On

Pure Hierarchy - SPICE	Pure	On	Off	Name	Name.Spice	Off	On	Off
Pure Hierarchy - General	Pure	Off	Off	Name	Name.Prefix	Off	Off	On

* Note: In Flat hierarchy mode, the Packaging option overrides the auto-naming feature in any case, so this setting is irrelevant. The Packager will place visible names on devices as they are placed.

In this table:

- Hier. Mode - refers to the Hierarchy Mode setting in the Design Preferences menu command.
- Dev. Auto Vis - refers to the Device Auto Naming - Visible option in the Design Preferences command. If this is On, the auto-assigned Name value will be displayed on the schematic.
- Packaging - refers to the Auto-Packaging Enabled menu command. If this is On, the device name and unit will be assigned automatically as each device is placed.
- Text Tool Sets - refers to the Text Tool Options item in Design Preferences. This determines which attribute field is set when the text tool is used to edit a device name on the schematic.
- Name Prefix Field - refers to the contents of the PrefixField design attribute field, set using the Set Design Attributes command. This determines where DesignWorks looks for a prefix to use in generating a device name.
- Unit, Part, Spice, Value Vis. - refer to the Visible By Default settings of the Unit, Part, Spice and Value attribute fields. These can be set using the Define Attribute Fields command. If this is On, the field value will be displayed on the schematic when the device is placed.

Default Pin Spacing

The lower group of buttons allows you to select from one of two commonly-used symbol styles supported by DesignWorks.

NOTE: This setting has no effect on the contents or availability of symbol libraries. It is up to you to choose the appropriate setting for the symbols you intend to use. The default DesignWorks installation installs only the DesignWorks-format symbols (1/8" pin spacing).

The following table summarizes the effect of these buttons.

DesignWorks (1/8")	This mode sets the default text sizes to 9-point for attribute and pin number text and the default pin spacing (i.e. for breakout symbols and DevEditor operations) to 2 grid units (approx. 1/8"). These settings will give best results with the DesignWorks-format symbol libraries provided with the package.
ANSI (1/4")	This mode sets the default text sizes to 12-point for attribute and pin number text and the default pin spacing to 4 grid units.* These settings will give correct text and line space proportions for ANSI-standard drawings when used with the ANSI or IEEE-format symbols provided. NOTE: The ANSI and IEEE symbols are not installed by the Easy Install option. You must select the Customize option to load them onto your disk.

* NOTE: The ANSI setting actually uses the setting specified in the PINSPACE keyword in the DesignWorks Setup file. If no PINSPACE item appears in the Setup File, the default value of 4 is used.

New Sub-Circuit Commands

A new sub-menu has been added under the Options menu in Schematic, entitled Sub-Circuit. Several of the commands in this menu are simply moved or renamed commands that existed in 3.0, but the "external sub-circuit" commands are new.

External Sub-Circuits

The concept of "external sub-circuits" is not new to this version, but it is much more visible due to the added menu commands. These commands were added primarily for use with the DesignWorks Simulator, although they also have other applications.

External vs. Internal Sub-Circuits

In order to simulate, analyze or generate reports from a hierarchical design, DesignWorks requires that all sub-circuits be loaded into memory. When a device symbol that represents a sub-circuit block is created in a library, the definition of it's sub-circuit can be associated with it in a number of ways:

- The sub-circuit can be stored with the symbol in the library, i.e. an "internal sub-circuit"

- The name and directory location of a circuit file can be stored with the symbol or added to the symbol after it is used in the schematic. The sub-circuit itself is not stored with the symbol, just the information required to locate it later. Schematic options allow the sub-circuit to be loaded automatically when the symbol is used, or on request at a later time. This is referred to as an "external sub-circuit".
- The symbol can be stored with no associated sub-circuit and one can be created or attached to it interactively after it has been placed in the schematic.

External References

The information used to locate an external sub-circuit is called an "external reference" and is stored in a number of attribute fields associated with the device. External references can be added to a device in various ways:

- When creating or editing a symbol in the DevEditor tool, an external reference is added by using the Sub-circuit and Part Type menu command and selecting the External Sub-Circuit option. This allows you to locate the external file, load the port definition and associate device pins with port connectors in the sub-circuit.
- If the Schematic tool is being used in conjunction with the DesignWorks Simulator, the SimLoad tool can be used to automatically locate and set external references for standard library parts as they are used or after they have been placed in a schematic.
- The Attach External Sub-circuit command in the Schematic tool can be used to associate an external sub-circuit with a symbol after it has been placed in a schematic.
- The external reference attributes can be entered manually. THIS IS NOT RECOMMENDED because any slight error in names or entry format will result in the file not being located.

See more information on sub-circuits and the SimLoad tool in the DesignWorks Simulator Reference Manual.

Creating External Sub-Circuits

External sub-circuits are simply normal DesignWorks design files with port connectors added to match the pins on the parent device. Following is a brief summary of the considerations affecting sub-circuits:

- In order to be loaded automatically using the SimLoad tool, the file must be named according to the conventions outlined in the DesignWorks Simulator manual.

• A port connector should be placed in the sub-circuit for each pin on the parent symbol, including no-connect pins. Failure to do this will result in the Schematic tool reporting an error each time the sub-circuit is opened and closed.

• Each port connector must be named to match the corresponding pin name on the parent device.

• The type of each port connector should match the type of the corresponding pin on the parent device. E.g. An input pin on the parent device should match with a "Port Conn In", or other appropriate type.

• The sub-circuit can contain any combination of other devices, including nested sub-circuits to any depth. The only restriction is that the sub-circuit may not contain the parent type or any other type that would create a recursive nesting of blocks.

Other considerations relating to the creation of sub-circuits are discussed in Chapter VII - Hierarchical Design in the DesignWorks/Schematic Reference Manual and in Chapter IV - Creating a Schematic for Simulation in the DesignWorks Simulator manual.

Sub-Circuit Menu Commands

IMPORTANT: When you attach or detach device sub-circuits, you are in effect changing the definition of the device. This always affects all other devices of the same type in the same design. This applies to all the sub-circuit commands following and any other sub-circuit editing operation in DesignWorks. If you want to affect only a single device, you must first use the Make Unique Type command in the Part Type sub-menu to isolate it from others of the same type.

Attach Sub-circuit

This is identical to the Attach Internal command described in the DesignWorks/Schematic Reference Manual. It allows you to choose a design that is already open in another window to be attached to the selected device symbol.

Refer to Chapter IX - Schematic Menu Reference for details.

Detach Sub-circuit

This is identical to the Detach Internal command described in the DesignWorks/Schematic Reference Manual. It detaches the sub-circuit that is displayed in the current circuit window from its parent device symbol. The sub-circuit will then be considered to be a completely separate design and the parent device symbol will be redefined as having no sub-circuit.

NOTE: Attach Sub-circuit works from the top down, i.e. you select the parent device symbol and then locate the sub-circuit to attach to it. Detach Sub-circuit works from the bottom up, i.e. the sub-circuit to be detached must already be open in the current window.

Refer to Chapter IX - Schematic Menu Reference for details.

Discard Sub-circuit

This new command discards any sub-circuit associated with the device that is selected on the schematic (and all others of the same type). It is similar to Detach Sub-circuit in that the parent device symbol will be redefined as being empty.

WARNING: The sub-circuit is completely lost and this operation cannot be undone!

Attach External Sub-circuit

This new command allows you to associate an external design file on disk with a device symbol in a schematic. It operates only on a single selected device at a time, but always affects all other devices of the same type in the design. It will optionally set the external reference attributes only, or set the external reference and immediately load and attach the sub-circuit.

The Attach External Sub-circuit command is also a convenient way of checking the status of an external reference. It locates the external file and compares the file's modified date with the parent device symbol. The "up to date" status and any problems locating the file will be displayed in the status box. If the selected device has an external reference which is currently out of date, clicking the OK button will reload it.

Update External Sub-circuits

This command allows you to scan all or part of a design and load or update external sub-circuits. This affects only devices that already have an external reference, created by one of the methods described previously. If the external file cannot be located, or if the external file is older or the same age as the last update, nothing is done. A summary box will be displayed after execution is complete showing counts of loaded and unlocated files.

NOTE: This command can dramatically increase the memory space occupied by your design.

Clear External Sub-circuits

This command allows you to scan all or part of a design and delete the sub-circuits from all devices having an external reference. This affects only devices that already have an external reference, created by one of the methods described previously. Devices with sub-circuit blocks that were not loaded using an external reference will not be affected.

This command is intended primarily to unload simulation models that were loaded using the SimLoad tool after they are no longer required. This can save considerable storage space for the design.

WARNING: This command discards a lot of information and cannot be Undone!

Auto-load External Sub-circuits

This command sets or clears the "Auto Load" option. It is disabled by default. When this option is enabled, each part is checked for an external reference when it is selected from a library. If there is a reference and the file can be located, it is loaded and attached immediately. Enabling this option has no effect on parts already placed in the design.

NOTE: If you select a part in the Parts palette that has already been used elsewhere in the design, the existing definition is used and no sub-circuit will be loaded.

External Reference Attribute Fields

The external reference information is stored in a number of attribute fields associated with the device, as summarized in the following table. Manual

editing of these fields is not recommended, since any slight spelling or format error will result in the external file not being loaded.

ExtCctName	This field contains the name of the design file containing the external sub-circuit.
ExtCctPath	This field contains the "path name" of nested directories leading to the file. The first item in the string is always the disk name, followed by a colon, followed by the names of nested folders, separated by colons.
ExtCctDate	This field contains an integer value expressed in decimal representing 1/60's of a second since Jan. 1, 1904. If this field is empty, the reference will be considered to be out of date.

New Setup File Options

The new setup file options described in this section provide the ability to set default options in new designs. They are not required in normal usage. See the DesignWorks 3.0 User's Manual for more information on editing the setup file.

Adding Default Attribute Field Definitions

The setup file keyword ATTRFIELD allows you to specify attribute field definitions that will be included in every new design that is created. The format of this keyword is as follows:

ATTRFIELD *fieldName*;

or

ATTRFIELD fieldName/options;

"fieldName" can be any valid attribute field name, i.e. consisting of letters, digits and the period and underscore characters "." and "_". The maximum field name length is 16 characters and spaces are not allowed.

If no options are specified, the field will be a device field with default maximum length and switch settings.

A number of switch setting options can be specified as letters and digits after a "/" following the field name. If any digits follow the "/", they are taken to be the maximum data length for this field. The default is 32000.

Any combination of other option letters can be selected from the following table. The order of the option letters is not important.

Any number of new fields can be added by using multiple ATTRFIELD lines. Any line specifying the name of an existing field will be ignored.

Option Letter	Meaning
A	Allow Carriage Return character in data
C	Allowed in design
D	Allowed in devices
F	Fixed, i.e. user cannot remove field definition
G	Group with Name
I	Instance data, i.e. can have different value inside different instances of the same hierarchical block
L	Location fixed on schematic
N	Only allow name characters in field value (NOTE: This is not enforced in the current version of DesignWorks)
P	Allowed in pins
R	Rotate with object
S	Allowed in signals
T	Do not display value list, even if a .List field is present
U	Only allow numeric characters in field value (NOTE: This is not enforced in the current version of DesignWorks)
V	Visible by default
W	Show Field Name by default
X	Read only value
Y	In Primary list by default

ATTRFIELD examples:

```
ATTRFIELD  Aperture/SY10;
ATTRFIELD  InputDelay/PYU10;
ATTRFIELD ModelText;
```

Setting New Design Command Action

The NEWDESIGN setup file keyword allows you to control the function of the New Design menu command. If this keyword does not appear in the

setup file, the New Design command will display the mode selection box described earlier in these notes.

NOTE: If you hold the Option key pressed on the keyboard while selecting the New Design command, the New Design mode box will be displayed regardless of the settings in the NEWDESIGN setup file keyword.

The NEWDESIGN keyword can be followed by arguments in one of the following two forms:

```
NEWDESIGN designMode pinMode;
```

or

```
NEWDESIGN fileName;
```

In the first form, NEWDESIGN is followed by two integers, separated by blanks. These numbers in effect select the option buttons in the New Design mode selection dialog. The first integer can be from 1 to 7 and selects one of the modes described in the New Design command above in the order that the buttons appear in the box. I.e. 1 = Flat - PCB and 7 = Pure Hierarchy - General. The second integer selects the pin space mode and can be 1 (DesignWorks 1/8") or 2 (ANSI 1/4").

The second form of the NEWDESIGN keyword allows you to specify the name of a file to open when the program is started or the New Design command is selected. This file can be an empty template containing your standard attribute fields and sheet settings. The format for the file name is the same as described for the DESIGN keyword in the DesignWorks User's Guide. This keyword is equivalent to performing an Open on the file except that the window will be titled "Designxxx" and the Save command will be disabled to avoid accidentally overwriting the template file.

NOTES: The name of the file specified after NEWDESIGN cannot start with a digit or the first form will be assumed.

NEWDESIGN examples:

```
NEWDESIGN 2 1; { Flat - SPICE }
NEWDESIGN :Template Files:My Startup Design;
```

Disabling Untitled Design at Startup

A new setup file keyword controls the creation of a new, untitled design when the program is first started. If the NOUNTITLED keyword appears in the file, the program will start up and just display its menus, with no default window opened. This entry will appear as follows:

NOUNTITLED;

Fine-Tuning Pin Number Text Display

The PINTEXT keyword allows you to adjust the display position of pin numbers on devices.

> IMPORTANT: This adjustment should not be required in normal use and should be used with caution. No checking is done on the range of these settings.

The format of this keyword is as follows:

PINTEXT *offsetH offsetV*;

"offsetH" and "offsetY" are positive or negative integer offsets which adjust the text position by the given amount in screen dots at Normal Size screen scale. All devices in the design are equally affected.

> NOTE: Changing these numbers in the setup file will not automatically recalculate the positions of pin numbers in existing designs. You can force a recalculate by using the Design Preferences command to change the pin text font or size, then change it back to the original setting.

Disabling "Loose End" Markers on Signal Lines

The NOLOOSEENDS setup file keyword disables the cross-markers that are normally displayed on the screen at the ends of unconnected line segments. The format of the command is:

NOLOOSEENDS;

Physical Hierarchy Mode Changes

A number of small refinements have been made in the operation of the hierarchy mode setting and device naming and packaging features. These

changes are primarily to simplify the creation of hierarchical PCB designs. These changes include:

• If the hierarchy mode is changed in the Design Preferences command, an option box is now displayed with options and suggestions for reformatting the design. For example, if switching between Flat and Physical mode, the device Name field should be copied to the InstName field in most cases.

• If the InstName field is changed and Name has no value, or has a default value and is invisible, then copy the new InstName value to Name. This ensures a minimum of surprises when reference is made to the Name field by a netlist or other means.

• If Name is not visible, use Name.Pt to display InstName field. The standard libraries provided with DesignWorks specify a default position for the Name field in the Name.Pt field. However, in Physical mode, the device name is usually stored in InstName. This change ensures that names are positioned as expected.

• Improvements in auto-positioning of attributes, especially for grouped and rotated items

• The Packager is now used to perform automatic name assignment. This results in more "intelligent" generation of names.

• On a Paste or Duplicate, new names are assigned to devices even if they already have them.

• A new text tool option allows you to select when the pencil tool sets the Name or InstName attribute field. In Physical Hierarchy designs intended for PCB use, the device name is usually stored in the InstName field. This option can be set using the Design Preferences command.

• To provide visual feedback on which attribute field is being edited, the name of the current field is shown in the status box at the lower left of the schematic window.

• Simulation I/O devices are considered to be pseudo-devs, i.e. not named automatically.

Report Enhancements

Several new report form keywords have been added to address specific netlist applications.

$COUNTINST - after $DEVICES or $SIGNALS

COUNTINST works exactly like $COUNT except that it counts the number of devices with different InstName values, instead of Name values.

$LEXICAL/$NUMERIC - After $SORT

These keywords control the way items are sorted in a netlist. The default method, which matches all previous DesignWorks versions, is $NUMERIC. In this mode, any numeric portion of a text string is extracted and treated as an integer. For example, this means that U10 will appear after U9 in a component list. The disadvantage of this method is that U010 and U10 will be treated as the same name, or component values .001 and .01 will be combined (because "001" and "01" have the same integer value). The $LEXICAL option performs a purely lexical sort, i.e. by ASCII code. E.g. U1, U11, U100, U2, U22, U3, etc. This is intended when sorting by component values or other random data, but will not normally give the desired results for names.

The following example shows the usage of this keyword.

$SORT $DEVICES &Part $LEXICAL &Value

This example will sort devices by Part field using the default numerical method, then sort any devices with the same Part value by the string in the Value field using the lexical method.

$REASSIGN - after $ASSIGNNAMES or $ASSIGNINSTNAMES

ASSIGNNAMES and ASSIGNINSTNAMES normally assign default names only to signals or devices that were unnamed, i.e. no name has been assigned by the user, the auto-naming features in Schematic or by any previous report.

However, in some cases, especially hierarchical designs, it is likely that duplicate names exists due to duplication of hierarchical blocks. For this reason, the $REASSIGN keyword has been added. When it appears anywhere after an ASSIGNNAMES and ASSIGNINSTNAMES, all names

that were auto-assigned by either the Schematic or the Report tool and have not been edited by the user will be reassigned a unique name.

The $REASSIGN keyword has been added to all ASSIGNINSTNAME commands in the standard report forms provided with DesignWorks. It will normally not be required in Flat or Pure mode designs.

$ASSIGNINSTNAMES

This works exactly like $ASSIGNNAMES except that it sets the InstName attribute field instead of the Name field. This is used for creating netlists from designs created in Physical Hierarchy mode.

Library Enhancements

New Symbol Format

Due to popular demand, a set of compact component symbols has been added to the DesignWorks libraries. These symbols do not meet ANSI drawing standards, but allow much more schematic to be placed on a sheet. These symbols are now the default installation. See the Installation section earlier in these notes for more information.

3.0 Manual Errata

Selecting Background Objects

Page 83 in the DesignWorks/Schematic Reference Manual indicates that the Command and Shift keys are used to select background objects. This should read Command and Option.

Index to DesignWorks 3.1.1 Release and Installation Notes

ASSIGNINSTNAMES report keyword 26, 27
ASSIGNNAMES report keyword 26
Attach External Sub-circuit command 17, 19
Attach Sub-circuit command 18
ATTRFIELD setup file keyword 21
Auto-load External Sub-circuits command 20
background objects 27
Clear External Sub-circuits command 20
converting 9
COUNTINST report keyword 26
Detach Sub-circuit command 19
DevEditor tool 17
devices
    name position 12
Discard Sub-circuit command 19
ExtCctDate attribute field 21
ExtCctName attribute field 21
ExtCctPath attribute field 21
external sub-circuits 16
    creating 17
flat hierarchy mode 13
hierarchy mode 12, 24
installation 3
InstName attribute field 14, 26, 27
LEXICAL report keyword 26
libraries 27
    installation 4
Name attribute field 14, 26
Name.Prefix attribute field 14
Name.Pt attribute field 12, 25
names
    devices 12
    port connector 18
New Design command 22
NEWDESIGN setup file keyword 22
NOLOOSEENDS setup file keyword 24
NOUNTITLED setup file keyword 24
NUMERIC report keyword 26
Packager tool 25
physical hierarchy mode 13, 24
pin spacing 15
PINTEXT setup file keyword 24
port connector 17, 18
pure hierarchy mode 14
REASSIGN report keyword 26
Report tool 26
    assigning names 26
    sorting 26
serial number 6
setup file 21
SimLoad tool 17
SORT report keyword 26
SPICE 13
sub-circuits 16
    external 16
    internal 16
symbol libraries 27
Update External Sub-circuits command 20

Copy Protection

For your convenience, the DesignWorks disk is not copy protected in any way, allowing you to copy the program onto your hard disk or backup floppies as desired. This does **not** imply that you are allowed to make copies for use on more than one computer, in fact you are specifically prohibited from doing so by your license agreement. By respecting that agreement, you will help us continue to bring you useful software products at reasonable prices.

What you have received

This package contains the 3.1.1 release of DesignWorks, and should contain the following:

1 Binder
1 License Agreement
1 Release Notes
7 Program Diskettes
1 Reference Manual Package
1 Registration Card
1 Serial Number Label or Sheet
1 Support Plan Enrollment Card

NOTE: Some of the above items will not be included in upgrade packages.

Installation Notes

> *See also: Chapter II - Getting Started in the DesignWorks/Schematic manual.*

In order to simplify installation and reduce the number of diskettes required to distribute DesignWorks, the release files are compressed onto a small number of diskettes using the Apple Installer and the StuffIt™ utility.

The Installer allows you to install a complete basic system in one operation, or to choose specific groups of items to be installed.